Unicenter

NetMaster Network Management for SNA Administrator Guide

1st edition

P01- 166

Booklist

Computer Associates
The Software That Manages eBusiness

Edition	Publication Number	Product Version	Min. MS Level	Publish Date
5th Edition	P01-166	4.0	MS 5.0	December 2001

The manufacturer of this documentation is Computer Associates International, Inc.

Table of Contents

Part I Introducing NetMaster for SNA

Chapter 1 About the Product 1-1

What Is NetMaster for SNA? 1-2
What Is NEWS? 1-3
NEWS Features and Benefits.. 1-3
What Is NTS?.. 1-4
NTS Features and Benefits 1-4
What Is NCS?.. 1-5
NCS Features and Benefits... 1-5
What Is NCPView?.. 1-5
NCPView Features and Benefits 1-6
What Is the Remote Operator Facility?.. 1-6
What Is the Network Management Facility? 1-6
What Is the SYSCMD Facility?... 1-7
What Is the NetView Operator Command Emulation Facility? 1-7
What Other Functions Are Available? 1-7
LAN Management.. 1-7
Information Database... 1-8
Activity Log 1-8
Session Replay 1-8
Command Entry 1-8

System Requirements for NetMaster for SNA 1-9

License Manager Program (LMP) Key Requirements 1-9

Chapter 2 How NEWS Works 2-1

Data Available to NEWS 2-2
- Unsolicited Data 2-2
- Solicited Data 2-2
- Response Time Data 2-2

How NEWS Obtains Data 2-3
- SNA Networks and the CNM Interface 2-3
- APPN Networks and SNA Management Services (SNAMS) 2-4
  - SNA Management Services Units 2-4
  - SNAMS Data Transfer 2-4
  - The &SNAMS Verb 2-4
- NEWS and Inter-System Routing (ISR) 2-5
- NEWS and PPI 2-5

NEWS Processing Concepts 2-5
- Network Services Control File 2-5
- Events 2-6
  - Event Types 2-6
  - Event Characteristics 2-7
  - Event Filtering 2-7
  - Resource Masks 2-8
  - Alert Monitor 2-8
- CNMPROC Record Processing 2-8

NEWS Facilities 2-10
- NCL Verbs and Procedures 2-10
- Unattended Solicitation 2-11
- NEWS Commands 2-11
- Alias Name Translation Facility 2-12

Reviewing and Reporting on Data 2-13

Chapter 3 How NTS Works 3-1

Data Available to NTS 3-2
Session Awareness (SAW) Data 3-2
Composition 3-2
Session Trace Data 3-3
Composition 3-3
Response Time Monitoring (RTM) Data 3-3
Composition 3-4
Route Configuration Data 3-4
Composition 3-4

How NTS Obtains Data 3-4
From VTAM Interfaces 3-5
Through Inter-system Routing Between NTS Systems 3-5
Single Image Presentation 3-5
Through MAI Sessions 3-6

Network Definitions and Names Used by NTS 3-6

System Resource Utilization 3-7

Session Start Processing 3-7

Output Processing 3-7
Session End Processing 3-8
NTS User Exit Processing 3-8

System Event Generation 3-8

NTS Database 3-9
Session Keep Counts and Database Slots 3-9
Connecting to the NTS Database 3-9

MAI Support 3-10
What Is an MAI Session? 3-10
The MAI/NTS Interface 3-11
Logging MAI Sessions to the NTS Database 3-12
Logging MAI Sessions to the NTS User Exit 3-12

Use of ISR Links 3-12
Session Configuration Data Sharing 3-12
NTS Single Image 3-13

Chapter 4 How NCS Works 4-1

How NCS Works 4-2
Access to Configuration Data 4-2

Chapter 5 How NCPView Works 5-1

How NCPView Works 5-2
NCPView and Connected Domains 5-2

Part II Administering NetMaster for SNA

Chapter 6 Administering NetMaster for SNA 6-1

Performing Administrative Tasks.. 6-2

Implementing Features 6-2
Implementing the Network Services Control File (NSCNTL) 6-3
Implementing NEWS Databases.. 6-3
Implementing the NTS Log Database.. 6-4
Identifying Your Initialization Procedure (SNAINIT) 6-5
Implementing NSCNTL Cache Options 6-5
Implementing NEWS Database Logging Options (NEWSDBOPTS) 6-6
Examples.. 6-6
Implementing Event Filters (CNMFILTERS) 6-7
Examples.. 6-7
Implementing Performance Objectives for Event Recording (CNMPERFOBJ) 6-8
Implementing SMF Event Recording Options (SMFT37)............ 6-8
Implementing the PPI Receiver (PPINETVALRT)...................... 6-9
Implementing Device Support Diagnostics (DEVICESUPP)........ 6-9

Customizing Facilities 6-10
Customizing NEWS Facilities 6-10
Customizing NTS Facilities.. 6-10
Customizing NCPView Facilities 6-10
Customizing NCS Facilities.. 6-11

Tailoring the NetMaster for SNA Startup Procedure 6-11
Enabling NetView Operator Command Emulation 6-11
Setting Up NTS 6-12

About Security 6-12
Security Considerations for Existing Users 6-13
Checking Existing User Group Definitions............................. 6-13
Customizing Existing Background User Definitions................ 6-14

Defining User Exits 6-15

Chapter 7 Tailoring NEWS 7-1

Reviewing Parameters to Send and Receive CNM Data...................... 7-2
Tailoring NEWS Database Options 7-2
Reviewing NCP Parameters and Operations 7-2
Customizing Device Configuration 7-3
LPDA Support.. 7-3
RTM Support 7-3
FCS Support... 7-4

Chapter 8 Maintaining the NEWS Database 8-1

Maintaining the NEWS Database.. 8-2
Enabling and Disabling CNMPROC Logging Options 8-2
Accessing Database Maintenance 8-3
Deleting Records Generically by Date and/or Node..................... 8-4
Deleting All Records... 8-4
Reorganizing the NEWS Database 8-5
Releasing the NEWSFILE from CNMPROC.......................... 8-5
Manually Reorganizing the Database 8-6

Chapter 9 Tailoring NTS 9-1

Defining NTS Classes.. 9-2
Specifying the DEFCLASS Command... 9-2
Defining Session Classes 9-3
Defining Resource Classes.. 9-5
Defining SAW Classes.. 9-7
Defining RTM Classes.. 9-9
Setting NTS System Parameters.. 9-11
Specifying the SYSPARMS Command 9-11
Defining the NTS ACB Name 9-13
Collecting NTS Session Accounting Data 9-13
Selective Accounting 9-13
Global Accounting.. 9-13
Collecting NTS Resource Statistics 9-14
Rules Governing Statistics Collection 9-15
Monitoring NTS Resource Availability................................... 9-15
Logging of Active Sessions at Shutdown 9-15
Enabling Intensive Message Recording 9-16
Enabling MAI Sessions... 9-16
Enabling NTS Session Event Generation 9-17

Setting the Data Correlation Interval 9-17
Setting Trace Limits 9-17
Setting Session Keep Counts 9-18
Setting VTAM Session and Trace Data Buffer Allocations 9-18
Default Allocations 9-18
Setting Resource Statistics Collection Intervals 9-19

Establishing Inter-System Routing Links 9-20
Configuring Your Systems for NTS-SI 9-20
Linking SNA Sessions 9-20
Distributing MAI Data Across ISR 9-20

Chapter 10 Collecting NTS Data 10-1

Opening the VTAM CNM Interface 10-2

Enabling NTS Session Awareness 10-2
Starting and Stopping Session Awareness Processing 10-2

Connecting and Disconnecting the NTS Database 10-3
Connecting the NTS Database 10-3
Disconnecting the NTS Database 10-3

Chapter 11 Maintaining NTS 11-1

Modifying NTS Processing 11-2
Modifying Processing While Session Awareness Is Inactive 11-2
Modifying Processing for Active Sessions 11-2
Using the NTSMOD Command 11-3

Modifying NTS Class Definitions 11-4
Showing the Current Values of Parameters and Classes 11-4
Making the Required Changes 11-4

Limiting NTS Trace Activity 11-5
Limiting the Number of Concurrent Traces 11-5
Limiting Trace PIU Collection 11-5

Maintaining the NTS Database 11-6
Modifying the Database Session Keep Counts 11-6

Writing NTS Records to SMF for Further Processing 11-7

Controlling the MAI/NTS Interface 11-7

Chapter 12 Tailoring NCS.. 12-1

Limiting the Size of Display Lists and Individual Node Displays 12-2
Limiting the Number of Lines in NCS Display Lists 12-2
Limiting the Number of Subnodes Displayed For a Node............ 12-2
Implementing NCS Display Limits.. 12-2

Chapter 13 Tailoring and Controlling NCPView................. 13-1

About NCP Monitoring 13-2
System Images 13-2
NCP Definitions 13-2
Working with NCPs 13-3
Monitoring Resources in a Multisystem Environment 13-3

Defining a System Image... 13-3

Defining NCP Resources... 13-4

Tailoring the NCPView NCL Exit 13-5

Chapter 14 Tailoring NetView Operator Command Emulation... 14-1

Modifying Table Entries.. 14-2
Considerations When Modifying Table Entries............................ 14-2

Chapter 15 Advanced Configuration Tasks 15-1

Implementing a Trouble Ticket Interface 15-2
Defining a Trouble Ticket Interface.. 15-2
Defining an E-mail Trouble Ticket Interface........................... 15-3
Defining a Custom Trouble Ticket Interface 15-3
Setting Up the Trouble Ticket Data Definition......................... 15-4

Implementing the Alert History Function.. 15-5
Reorganizing Files and Monitoring Space Usage.................... 15-6

Applying Alert Monitor Filtering 15-6

Forwarding Alerts... 15-7
Implementation 15-7
The SNMP Trap Definition.. 15-8
$AMEVFWD 15-8

Destination Details 15-8
Forwarding Alerts to Unicenter Event Management 15-9
Sending Alerts as SNMP Traps 15-10
Sending Alerts as NMVTs 15-11
Forwarding to NetMaster for SNA 15-11
Forwarding to NetView 15-12
Filtering Details 15-12

Enabling Multi-system Support 15-13
Defining Communications for NEWS and for NTS 15-14
Example 15-14
Defining ISR Inbound Parameters (ISRIN) 15-14
Defining ISR Outbound Parameters (ISROUT) 15-14
ISR and NTS 15-15
ISR Environments 15-15

Managing Focal Points 15-15
About Focal Points and Entry Points 15-15
About Local and Backup Focal Points 15-16
About Nesting Focal Points 15-16
Browsing, Updating, or Deleting Focal Points 15-17
Defining Backup Focal Points 15-18
Defining Nesting Focal Points 15-19

Managing Entry Points 15-19
Activating a Focal Point 15-20
Maintaining Entry Point Definitions 15-20
Updating an Entry Point 15-22
Defining an Entry Point 15-22
Deleting an Entry Point 15-22

Maintaining Control File Records 15-23
Accessing the Control file Category Maintenance Panel 15-23
Browsing, Modifying, Deleting, or Adding Control Records .. 15-24

Maintaining Resource Alias Names 15-26
About Alias Name Translation 15-26
Example of Resource Alias Name Translation 15-27
Example of Class-of-Service or Logmode Alias Names Translation 15-27
Displaying Alias Name Definitions 15-28
Example of Displaying Alias Names 15-28
Defining Alias Names 15-29
Example of Defining an Alias Name 15-29

Defining Generic Names 15-29

Defining Generic Alias Names 15-30

Defining Generic Network Names... 15-30

Replacing Alias Names 15-30

Example of Replacing an Alias Name 15-30

Deleting Alias Names 15-31

Example of Deleting an Alias Name 15-32

Testing Alias Names Translation 15-32

Examples of Testing Translation 15-32

Allocating NCP Unformatted Dumps.. 15-34

Example 15-34

Expected Unformatted Dump File Characteristics 15-35

Estimating Storage Requirements for Processing NCP Dumps Using NCPView.. 15-35

Integrating NCS with a Configuration Management Database 15-36

About Exit $NCCNFG 15-36

When $NCCNFG Is Used... 15-36

Integrating with INFO/MASTER 15-37

Integrating with Any Other Source 15-38

Creating User Alerts 15-38

Uses for Alerts.. 15-38

Sending Alerts from NCL Procedures 15-39

Using the $NWALERT Procedure 15-39

Creating Alerts 15-40

Accessing the NEWS Create an Alert Menu 15-40

Creating an Operator Alert 15-41

Creating a Non-generic (Basic) Alert 15-41

Creating a Generic Alert 15-44

Part III Reference

Appendix A Security Exit Support Requirements................ A-1

Structured Field Descriptions A-2

Appendix B About the CNM Interface B-1

CNM Application B-1
Network Services Request Units (NS RUs).................................. B-2

CNM Data from Network Resources B-2
Forward Request Units... B-3
Deliver Request Units B-3
NMVT NS Request Units B-4
REQMS NS Request Units B-5
Defined REQMS Data Types.. B-5
RECFMS NS Request Units B-6
Defined RECFMS Data Types.. B-6
RECMS NS Request Units.. B-7

SSCP Related CNM Requests B-7
Translate Inquiry and Reply Request Units B-8
CNM-RU... B-8

How Records Are Processed B-8
Record Type Recognition B-8
Processing Code Assignment.. B-9
NMVT Records and MDS-MUs... B-9
RECFMS Records.. B-9
RECMS Records.. B-9
Processing Path Selection B-10
Process Path Definitions B-10
Solicited CNM Record Processing.. B-10
Unsolicited CNM Record Processing B-11

References... B-11

Appendix C About the Session Awareness (SAW) Interface C-1

Using NTS Classes C-1
SAW Classes C-2
RTM Classes C-2
Session Classes C-3
Session Classification C-4
RTM Class Processing C-5
Resource Classification C-6
Collecting Resource Statistics C-7
Collection Intervals C-7
Resource RTM Statistics C-8
Cross-Domain Statistics C-8
Monitoring Resource Availability C-9
NTS Resource Statistics Logging C-9
Collecting Further Data C-10
Session and Resource Data C-10
Session Trace Data C-10
Resource Trace Request C-11
Accounting Data C-11
Selective Collection of Accounting Data C-11
Global Collection of Accounting Data C-12
Response Time Monitor Data C-12
Soliciting RTM Data C-13
Analyzing RTM Data C-13
Data Correlation C-13
Enforcing the Correlation Interval C-14
How NTS-SI Works C-14
NTS-SI Configuration C-15
Example C-15
Data Propagation C-16
Star Network Configuration C-17
How NTS Systems Share Data C-18
Reference Network Concept C-18
Dormant NTS Concept C-18
SAW Data Sharing C-18
SAW Data Sharing Rules C-19
SAW Data Clean-up C-19
Session Data Sharing C-20
Necessary Conditions C-21
Session Data Sharing Rules C-21

Session Data Flows C-22
Session Data Availability Notifications C-22
Session Data Solicitation C-23
Unsolicited Session Data at Session End C-23

Appendix D NEWS Device Solicitation Procedures D-1

Line Command Procedures D-2

$NW386SO D-3

$NWDS13B D-4

$NWFCSSO D-8

$NWLPDA2 .... D-9

$NWRTMSO D-12

$NWRUNCM D-14

$NWSOLCT D-16

$NWVPDSO D-18

Appendix E Implementing the NEWS User Exit E-1

Sample NEWS Exits E-2

How the NEWS Exit Is Called E-2
Processing the NEWS Exit E-3

NEWS Exit Coding Requirements E-3
Maintaining Registers on Entry to an Exit E-3
Parameter List Format E-4

About Exit Function Codes E-5
Function Code 0 .... E-5
Function Code 4 .... E-6
Translate-Inquiry RUs E-7
Function Code 8 .... E-7

Issuing Messages from the NEWS Exit E-7

NEWS SMF Record Formats E-8
SMF Header Section E-9
CNM Record Section E-10
TARA Header Section E-10
TARA Data Section E-11

Appendix F Implementing the NTS User Exit F-1

Sample NTS Exit F-1

How the NTS Exit Is Called F-2

Processing the NTS Exit F-2

NTS Exit Coding Requirements F-3

Maintaining Registers on Entry to an Exit F-3

Parameter List Format F-3

About Exit Function Codes F-4

Function Code 0 .... F-4

Function Code 4 .... F-5

Function Code 8 .... F-7

Issuing Messages from the NTS Exit F-7

Appendix G NTS SMF Record Formats G-1

NTS SMF Record Description for All Sub-types G-2

SMF Header Section G-2

Data Section G-2

Product Section G-2

NTS SMF Record Sub-type 1 to 7 Description G-2

Session Configuration Section G-2

Session Accounting Section G-3

Session RTM Section G-3

Explicit Route Section G-3

NTS SMF Record Sub-type 255 Description G-3

Resource Configuration Section G-3

Resource Accounting Section G-3

Resource RTM Data Section G-3

SMF Header Section G-4

Data Section G-5

Product Section G-6

Session Configuration Section G-7

Session Accounting Section G-9

Session Response Time Measurement Section G-10

Session Route Configuration Section G-11

Resource Configuration Section G-12

Resource Accounting Section G-13
Resource Response Time Measurement Section G-14

Appendix H NTS SNA Descriptor Table H-1

Macro Syntax H-2
$NTRUDEF H-2
$NTSCDEF H-2
$NTFMHDF H-3

Table Formats H-3

Macro Compile Errors H-4
$NTRUDEF H-4
$NTSCDEF H-4
$NTFMHDF H-4

Table Modification Procedure H-5

Appendix I NTS Storage Estimates I-1

Active Session Data I-1
When Using NTS-SI I-2

History Session Data I-2

A Database Initialization Strategy I-3

Glossary

Index

Figures and Tables

Figure 2-1. How CNMPROC Processes Data from the SNA Network 2-9
Figure 3-1. MAI Session.. 3-11
Figure 3-2. Transfer of Session Data via an ISR Link 3-13
Figure 4-1. Transfer of NCS Data Across an INMC Link 4-2
Figure 8-1. NEWS : Database Maintenance Menu 8-3
Figure 15-1. Alert Monitor : Email a Trouble Ticket Panel......................... 15-3
Figure 15-2. Alert Monitor : Custom Trouble Ticket Panel......................... 15-4
Figure 15-3. Alert Monitor : Trouble Ticket Data Entry Definition Panel ... 15-5
Figure 15-4. Alert Monitor : Filter Definition Panel.................................... 15-6
Figure 15-5. Backup Focal Point.. 15-16
Figure 15-6. Nesting Focal Point.. 15-17
Figure 15-7. SNA : Focal Point Administration Menu................................ 15-17
Figure 15-8. NEWS : SNAMS Backup Focal Point Definition Panel 15-18
Figure 15-9. NEWS : SNAMS Nesting Focal Point Definition Panel......... 15-19
Figure 15-10. SNA : Entry Point Administration Menu................................ 15-20
Figure 15-11. NEWS : Entry Point Definitions Panel—Update Mode.......... 15-21
Figure 15-12. NEWS : SNAMS EP Definitions Panel—Update Mode......... 15-22
Figure 15-13. SNA : Control File Administration Menu 15-23
Figure 15-14. NEWS : Category Selection Panel... 15-24
Figure 15-15. NEWS : Generic Alert Descriptions Panel 15-25
Figure 15-16. Example of Alias Name Translation...................................... 15-27
Figure 15-17. SHOW DEFALIAS Results.. 15-28
Figure 15-18. Replacing an Alias Name... 15-30
Figure 15-19. Replaced Alias Name Results.. 15-31
Figure 15-20. Deleting an Alias Name... 15-32
Figure 15-21. Testing Alias Name Translation 15-33

Figure 15-22. Alias Name Testing, Example 1 15-33
Figure 15-23. Alias Name Testing, Example 2 15-33
Figure 15-24. NCP : NCP Dump Menu.. 15-34
Figure 15-25. NEWS : Create an Alert Menu 15-40
Figure 15-26. NEWS : Create a Basic (Non-generic) Alert Panel 15-42
Figure 15-27. NEWS : Create a Generic Alert Panel 15-44
Figure B-1. Data Exchange Between VTAM and CNM APPL.................. B-2
Figure B-2. Sessions Used by the Network Services RUs........................... B-4
Figure C-2. Capturing Trace Data.. C-12
Figure C-3. Transfer of Session Data via an ISR Link C-16
Figure C-4. Data Propagation Across ISR Links C-16
Figure C-5. Optimal NTS Configuration C-17
Figure C-6. Capturing Trace Data From Another Domain C-20
Figure E-1. Structure of NEWS User Exit Parameter List.......................... E-4
Figure F-1. Structure of NTS User Exit Parameter List F-4
Figure H-1. Example of SMP UCLIN Statements for a USERMOD H-5

Table 6-1. Implementing NetMaster for SNA Features............................ 6-2
Table 9-1. Valid Values for the DEFCLASS SESSION Command Operands 9-4
Table 9-2. Valid Values for the DEFCLASS RESOURCE Command Operands 9-6
Table 9-3. Valid Values for the DEFCLASS SAW Command Operands . 9-8
Table 9-4. Valid Values for the DEFCLASS RTM Command Operands . 9-10
Table 9-5. SYSPARMS Operands for Tailoring NTS 9-12
Table 15-1. Basic Alert Subvectors... 15-43
Table 15-2. Generic Alert Subvectors... 15-46
Table A-1. List of Structured Fields... A-2

Part I

Introducing NetMaster for SNA

1

About the Product

Unicenter NetMaster Network Management for SNA is a network management product that simplifies the processes involved in managing complex computer networks.

NetMaster for SNA can manage many thousands of network addressable units, resulting in better network performance and availability, and faster recovery from network errors.

This chapter discusses the following topics:

- What Is NetMaster for SNA?
- What Other Functions Are Available?
- System Requirements for NetMaster for SNA
- License Manager Program (LMP) Key Requirements

What Is NetMaster for SNA?

NetMaster for SNA provides functions that allow you to do the following tasks:

- Monitor and react to network errors
- Command and control network resources
- Track session information from a single console
- Monitor Network Control Programs (NCPs) across the network

NetMaster for SNA provides tools that collect network information and monitor network devices so that network operators can take action before problems occur. You also get information on network status changes and network sessions so you get a complete picture of network activity.

NetMaster for SNA handles commands, messages, responses, and alarm information from multiple systems simultaneously. It can process information from a wide range of SNA and non-SNA devices and applications.

NetMaster for SNA provides a single-image facility that lets you monitor any domain from any terminal in your network, without switching between domains and without terminating and re-establishing sessions. You need to look in only one place for the information you need.

NetMaster for SNA provides real-time session-level information as well as detailed session diagnostics. This access to session start times, stop times, number of bytes passed, session trace records, and response time statistics, allows you to track actual network usage and performance. For longer-term analysis, NetMaster for SNA maintains a database of session histories. This information is available to help you plan and configure your network, to ensure you have adequate resources where they are needed.

NetMaster for SNA comprises a menu-driven system of full-screen panels with context-sensitive online help.

You can tailor NetMaster for SNA components to suit your site requirements. The NetMaster for SNA components are:

- Network Error Warning System (NEWS)
- Network Tracking System (NTS)
- Network Control System (NCS)
- NCPView
- The Remote Operator Facility (ROF)
- The Network Management facility
- The SYSCMD facility
- The NetView Operator Command Emulation facility

What Is NEWS?

NEWS provides a centralized system for the continuous monitoring of network error conditions so network operators can detect signals of imminent hardware failures, and quickly recognize and isolate faults that occur.

NEWS recognizes and logs events (for example, degraded user response time through a particular controller). It provides the means for filtering events by service objectives and commitments. Selected event records can be stored in a database for later analysis. Other events may initiate procedures that result in operator attention messages, or in automatically-generated problem tickets.

NEWS Features and Benefits

To help you monitor your network, NEWS provides these facilities:

- Masks that you tailor to filter event records received, and to respond to the events as you require
- In event of a failure, diagnostics that help pinpoint the cause of the failure and speed up its reversal
- Amalgamation of various types of unsolicited data from SNA and non-SNA resources in the network, and issue of requests for specific information from VTAM and certain hardware components
- The alias name translation facility to translate resource names, to avoid any confusion should a duplicate resource name be encountered in another network. This applies, for certain releases of VTAM and NCP, when SNA Network Interconnection (SNI) is used.
- Enhanced session hierarchy displays and session partner information, if the Network Tracking System (NTS) is also installed
- The ability to issue of operating system commands and return the results.
- The ability to forward alerts to Unicenter TNG where they can be monitored from the TNG Event Management Console.

For more information about using NEWS, see the *Unicenter NetMaster Network Management for SNA User’s Guide*.

What Is NTS?

NTS provides improved session visibility to help you determine problems and analyze your network's performance. NTS obtains information about logical network connections from VTAM and other NTS systems.

NTS Features and Benefits

To help you manage your network and locate problems, NTS does these:

- Provides an integrated view of activity across multiple SNA domains and networks
- Accumulates traffic statistics for sessions and resources to allow monitoring of network performance
- Uses the data available to it to build a model of the networking environment in which it is executing
- Traces selected sessions to determine the presence of problems in a logical network

Other NTS benefits are:

- Writes selected session details to a database, to provide an historical record of network activity that you can analyze to determine patterns, and locate previous occurrences of a particular problem
- Interfaces with the Multiple Application Interface (MAI) component of the SOLVE:Access product, to provide you with end-to-end visibility of MAI virtual sessions
- Is tailorable, to enable the most efficient use of computer resources and meet the specific needs of your installation

For more information about using NTS, see the *Unicenter NetMaster Network Management for SNA User's Guide*.

What Is NCS?

NCS is an effective and easy-to-use system for displaying and controlling network resources that are defined in any domain in which NCS is running, or in any domain connected by an Inter-Management Services Connection (INMC) link.

NCS Features and Benefits

NCS enables you to display:

- Lists by resource type, in summary form
- Detailed, graphical representations of individual resources and their subordinate nodes
- NEWS events for a selected resource
- NTS active sessions for a selected resource
- SNA resource session status codes

NCS also enables you to:

- Activate and deactivate resources from selection lists
- Enter NCS options to issue VTAM display, modify, and vary commands
- Display and control resources in other VTAM domains, enabling central control of all network resources

For information about using NCS, see the *Unicenter NetMaster Network Management for SNA User's Guide*.

What Is NCPView?

NCPView is a NetMaster for SNA application that supports IBM 3745 and 3746-900 communications processors that run a Network Control Program (NCP). This support provides an increased level of visibility of the configuration, and problems occurring with these communication processors.

It enables the NetMaster for SNA user to monitor:

- Token-ring resources
- Buffer and central control unit (CCU) utilization
- Virtual routes and transmission groups
- Internet protocol (IP) resource statistics
- SNI connections
- Frame relay resources

NCPView enables network operators to be proactive in monitoring, troubleshooting, and balancing loads among the communications controllers that are running an NCP.

It does this by enabling operators to display these types of information:

- Particular aspects of an NCP, such as associated virtual routes, transmission groups, and control block pools
- NCPs in other domains
- Information derived from an unformatted NCP dump

You can analyze information contained in these NCPView displays and use it to aid problem diagnosis. For more information about using NCPView, see the *Unicenter NetMaster Network Management for SNA User's Guide*.

What Is the Remote Operator Facility?

NetMaster for SNA is designed for an integrated network of interconnected systems. The Remote Operator Facility (ROF) allows a central operator to display and control resources in any VTAM domain where there is an INMC link to a remote NetMaster system.

In order for the full capabilities of the NetMaster for SNA product to be realized on a remote system, two criteria must be met:

- The NetMaster for SNA product must be licensed on the remote system.
- Operator(s) must be defined to the security system on the remote system—that is, they must be defined with adequate authority. In addition, their command authority should allow them to issue VTAM commands.

For more information about ROF, see the *Management Services User's Guide.*

What Is the Network Management Facility?

The standard Network Management facility provides the use of the SPO and the PPO interfaces. The SPO interface is used to issue commands to VTAM. The PPO interface is used by the distributed NCL procedure PPOPROC to receive all important network messages, particularly to intercept unsolicited VTAM PPO messages.

This facility provides an additional source of information to NetMaster for SNA. It supports VTAM operator commands (such as D and F), and implements others as Management Services (MS) commands (such as TRACE and ACT).

What Is the SYSCMD Facility?

The SYSCMD facility gives you the ability to issue OS/390 operating system commands and receive responses without having to use a *real* operating system console.

For information about implementing the SYSCMD facility, see the *Management Services Administrator Guide*.

For information about using the SYSCMD facility, see the *Unicenter NetMaster Network Management for SNA User's Guide*.

What Is the NetView Operator Command Emulation Facility?

The NetView operator command emulation facility is provided by NetMaster for SNA to assist former NetView users with the commands used in NetMaster for SNA.

This allows users to operate NetMaster for SNA by using the same commands and procedures they are accustomed to using with NetView.

What Other Functions Are Available?

There are functions accessible from the Unicenter NetMaster : Primary Menu that are not components of NetMaster for SNA. These available functions are:

- LAN management
- Information database
- Activity log
- Session replay
- Command entry

LAN Management

NetMaster for SNA supports IBM LAN Management Support.

LAN Manager network, adapter, and bridge functions (including bridge configuration) can be executed from the host, if you are running the IBM LAN Manager and it supports these functions.

Information Database

The Information Database contains network and Management Services information, organized by categories such as:

- 3174 error codes
- Messages
- SNA sense codes
- SNA resource status codes

Some categories of information are distributed with Management Services. However, you can add your own installation-specific categories.

Activity Log

The Activity Log browse function provides a full-screen display of the Management Services log, which is a record of all system activities.

You can search through the activity log both forwards and backwards by time, date, or keyword. You can also select an alternative log format, or issue a command and see the result reported at the bottom of the log.

Session Replay

If you are licensed for SOLVE:Access, you have access to the Session Replay facility, which is an aid to problem diagnosis.

The Session Replay facility enables you to record all I/O activity associated with one or more terminals, by using MAI. You can then review the recorded activity frame by frame, or as a sequence.

Command Entry

The Command Entry facility provides you with a full command entry and response screen. This enables you to issue system commands and to view the results, which are displayed in the form of a scrollable list.

Commands can be sent to both your own and to any connected NetMaster systems. Any commands you issue are retained in a logical stack and can be retrieved.

System Requirements for NetMaster for SNA

To run NetMaster for SNA, the following system requirements apply:

- You must have Management Services and Automation Services installed on your system before you can run NetMaster for SNA Version 4.0. When you purchase NetMaster for SNA, you are supplied with the current version of both Management Services and Automation Services.
- The minimum maintenance level of both Management Services and Automation Services with which NetMaster for SNA Version 4.0 can operate is Version 5.0.
- If you want to use NetSpy SNA agents and collect NetSpy data, the minimum level of Unicenter NetSpy Network Performance that you need is Version 6.0.
- The minimum operating system level on which NetMaster for SNA Version 4.0 will operate is OS/390 V2R6 or later, including z/OS.
- The minimum VTAM level with which NetMaster for SNA Version 4.0 can operate is VTAM Version 4.2.

If you are migrating to NetMaster for SNA Version 4.0 from a previous version, see the *Unicenter NetMaster Network Management for SNA Version 4.0 Release and Migration Guide* for further information.

License Manager Program (LMP) Key Requirements

To be able to access and use any NetMaster for SNA components, your installation license must have the License Manager Program (LMP) keys for the relevant product names specified in the current system initialization PROD= parameter of the JCL.

For details of the PROD= parameter, with a table of product name keys and LMP codes, see the *JCL Parameters* appendix of the *Management Services Administrator Guide*.

For lists and detailed descriptions of the structured fields for each supported NetMaster for SNA component, and a brief description of the support, see Table A-1 on page A-2.

2

How NEWS Works

This chapter presents an overview of how the NEWS component of NetMaster for SNA works, and introduces concepts that you need to understand to gain the maximum benefit from using this component.

This chapter discusses the following topics:

- Data Available to NEWS
- How NEWS Obtains Data
- NEWS Processing Concepts
- NEWS Facilities
- Reviewing and Reporting on Data

Data Available to NEWS

NEWS processes a wide variety of data received from different sources, and responds to the data it receives in an appropriate (user-definable) way. The types of data are:

- Unsolicited data
- Solicited data

NEWS also processes solicited and unsolicited data from the 3x74 Response Time Monitor.

Unsolicited Data

Unsolicited data—which originates from network nodes and distributed devices—received by NEWS includes:

- Traffic statistics
- Temporary and permanent error statistics
- Errors detected by network nodes
- Alert data generated by network components

Solicited Data

As a management application, NEWS can solicit further data from network components by issuing requests via the CNM interface. The following types of data can be requested:

- Device-dependent error data
- Microcode level data
- Link error data recorded by network devices

Response Time Data

NEWS also supports the 3x74 Response Time Monitor (RTM). This enables NEWS to receive response-time data, both solicited and unsolicited, that is maintained by the 3x74. The RTM data can be displayed in either numeric or bar-graph formats. NEWS also enables you to change the status of the RTM component in a 3x74. For instance, you can start or stop monitoring, or change the response time limits.

If you are also licensed for the NTS component of NetMaster for SNA, you can obtain more detailed response time data. In particular, you can see whether your installation-specific response time objectives are being met.

How NEWS Obtains Data

NEWS, as part of the NetMaster for SNA product, is linked to VTAM and your SNA network by means of two Access-method Control Blocks (ACBs), through which specific types of data are channeled.

The ACBs are:

- The primary NetMaster for SNA ACB
- The Communications Network Management (CNM) ACB

NEWS obtains data from the following sources, via the specified routes:

Source	Route
The local SNA subarea network	Via the VTAM System Services Control Point (SSCP) and the CNM ACB
The Control Points (CPs) of APPN nodes	Via LU 6.2 sessions, which carry SNA management data, and the NetMaster for SNA primary ACB
Connected systems	Via a Management Services Inter-System Routing (ISR) connection
Applications running on the same host that use PPI	Via the program-to-program interface (PPI)

These forms of data delivery are discussed in the following sections.

SNA Networks and the CNM Interface

The CNM interface provides a means by which a suitably authorized CNM application (such as NEWS) can maintain a session with an SSCP, to acquire data from Physical Units (PUs) in your SNA network. This session is established when NEWS successfully opens its VTAM CNM ACB, allowing data to be exchanged with the SSCP of the VTAM under which NEWS is running.

NEWS receives the following types of data from the SSCP:

- Unsolicited, as a result of some network event
- Solicited, as a reply to a previous request for data issued by NEWS itself
- As a solicitation from the SSCP, requesting that NEWS send some data in response

NEWS sends data to the SSCP for the following reasons:

- To solicit data from a network resource
- In response to a solicitation request from the SSCP

More detailed information about the CNM interface can be found in Appendix B, *About the CNM Interface*.

APPN Networks and SNA Management Services (SNAMS)

When NetMaster for SNA starts up, a primary ACB linking it to VTAM is opened. During initialization, NEWS registers with Multiple Domain Support (MDS) as the ALERT-NETOP application that acts as focal point for alert collection from your APPN network.

MDS is a component of SNA Management Services (SNAMS) that facilitates the routing of management data between applications. It enables management roles—that is, which node is to be the focal point for the receipt of which data—to be negotiated between nodes. (See the IBM publication, *SNA Management Services Reference*, for a more comprehensive description of the functions and services associated with SNAMS.)

SNA Management Services Units

SNA data units are known as Management Services Units (MSUs). There are various types of SNA MSUs, and many reasons for the generation of each type. As a result, NEWS needs to apply rules to classify incoming records, to determine the relative importance of the data that each carries, and whether a response is merited.

SNAMS Data Transfer

The SNAMS architecture makes use of SNA Management Services transport which consists of a number of APPC transactions used to transport SNA MSUs across the network.

The &SNAMS Verb

The NetMaster for SNA &SNAMS verb enables NCL procedures to participate as management applications in the SNAMS architecture. The partner application can exist within the same NetMaster system, a remote NetMaster system, or any other system that supports MDS functions. (For more information on the &SNAMS verb, see the Management Services manual, *Network Control Language Reference*).

NEWS and Inter-System Routing (ISR)

The Inter-System Routing (ISR) feature of Management Services allows data to be transparently routed to remote processing environments in other systems.

NEWS takes advantage of ISR to exchange data with remote systems. Requests, including CNM requests, can be routed to remote NEWS systems for processing, and the results returned to the originating system. This can be used to change the operation of, or solicit data from, a device managed by VTAM in the remote system. (The NEWS Device Support facilities use this capability.) Distributed processing enables NEWS systems to be more specialized in their processing, while still maintaining the capability to process any type of event.

ISR also enables a NEWS system to deliver unsolicited records to a remote NEWS system for processing. This enables data received by one NEWS system to be propagated to other systems, where it might be logged, acted upon, or ignored.

By using ISR, you can also implement centralized management, resulting in one NEWS system processing all data received by NEWS in linked systems. This enables you to control all database logging from one system.

NEWS and PPI

Applications running on the same host can communicate by means of PPI, a support facility provided by either the NetMaster for SNA subsystem interface, or by NetView. Users of PPI can send various types of data to this interface, which then distributes the data to the application registered to receive that particular type of data.

NEWS registers with PPI as the receiver of generic alerts, in order to monitor—and, if necessary, report on—the state of other applications using PPI.

NEWS Processing Concepts

This section explains the various concepts relating to the processing of all records received by NEWS.

Network Services Control File

The processing requirements for records received by NEWS are determined by a control database called the Network Services Control File (NSCNTL). This database contains control codes that describe the processing to be performed for each type of record received by NEWS. It also acts as a data dictionary, and is used to interpret the control codes present in the record.

This approach of determining processing requirements through data held on a control database provides extensive flexibility. The NEWS Control Function menu provides functions that allow you to add support for non-standard devices and requirements, or modify a control record held on the database. Any alterations or additions made to the control database are effective immediately.

NSCNTL is used by CNMPROC, a specialized NCL procedure, to process all records received by NEWS (see the section, *CNMPROC Record Processing*, on page 2-8).

Events

Any record received by NEWS is classified as an *event*, or as having the capacity to generate an event. The concept of the event allows different types of data to be grouped into one broad category, to provide a chronological record for a network node.

Generally, unsolicited records notifying NEWS of network errors are immediately classified as events. Records carrying statistical data have the potential to generate an event, if they include values that exceed thresholds set by your installation.

Event Types

Although it is convenient to group various sources of network data under the events umbrella, additional information is required to assist processing. Each event record is classified as one of the following types:

- Permanent error (PERM)
- Temporary error (TEMP)
- Performance (PERF)
- Intervention required (INTV)
- Customer application (CUST)
- End user (USER)
- SNA summary (SNA)
- Intensive mode record (IMR)
- Availability (AVAL)
- Notification (NTFY)
- Environmental problem (ENV)
- Installation problem (INST)
- Operational or procedural error (PROC)
- Security (SCUR)
- Delayed recovery (DLRC)
- Permanently affected resource (PAFF)
- Impending problem (IMP)

- Bypassed (BYPS)
- Redundancy lost (RLST)
- Unknown (UNKN)

Each record contains a field that identifies the type of data in the record, and determines the event type.

Event Characteristics

The characteristics that identify an event are generally represented by one or more codes within the record. These codes might describe:

- The reason why the event was generated
- The probable cause or causes of the event
- The severity of the problem associated with the event
- The resources affected by this problem
- The recommended remedial action

The codes might also provide additional information that assists in determining the cause of the associated problem, and are used by NEWS, in conjunction with the control file, to determine record processing.

Event Filtering

It is recognized that different installations have different network configurations and employ a variety of device types, some of which have their own special requirements. It is also recognized that installations assign differing levels of importance to particular network events and problems in their environments.

NEWS provides for this situation by passing all event records through a process termed *event filtering*, using parameters set in the Control File. This process enables you to discard those event records not required by your installation, while bringing to the attention of the network operator those considered to be of great importance (and various options between these two extremes).

For each of the event types described on the previous page, you can filter by using resource masks. You can also set options that control how the event is recorded in the NEWS database and whether an alert is generated for display on the Alert Monitor.

Event filtering is performed by the NEWS and user CNM processing procedures. CNMPROC uses control values that you set through the CNMFILTERS parameter group in ICS to perform filtering. For information about how to set filters for event recording, see the section, *Implementing Event Filters (CNMFILTERS)*, on page 6-7.

Resource Masks

Because all records are sent by, or on behalf of, a network resource, it is possible to restrict the processing of certain types of data to particular resources.

Resource masks are defined generically, and are used to include or exclude records based on the originating resource name. Records that do not satisfy resource masking criteria are discarded.

Alert Monitor

Through NEWS, you can generate alerts to increase operator awareness of important events in the network. Any record received by NEWS can be classified, through event filtering, as an alert. The alert monitor contains the most recent alerts produced by NEWS. The display is updated as new alerts arrive, or the status of alerts on the display is changed.

CNMPROC Record Processing

CNMPROC is a special NCL procedure that acts as a focal point application for SNA management data. CNMPROC executes in a background environment under user ID *xxxx*CNMP, where *xxxx* is the four-letter Management Services domain ID. A working version of CNMPROC is distributed as $NWCNMPR.

The function of CNMPROC is to:

- Analyze the content of each record delivered to it, in conjunction with the Network Services Control File (NSCNTL), to determine the processing requirements
- Process the record accordingly

CNMPROC is written as a continuous procedure (like PPOPROC and LOGPROC) and uses the &CNMREAD verb to receive each record as it becomes available for focal point processing. CNMPROC does some pre-processing to identify the record and then calls other procedures to perform further processing.

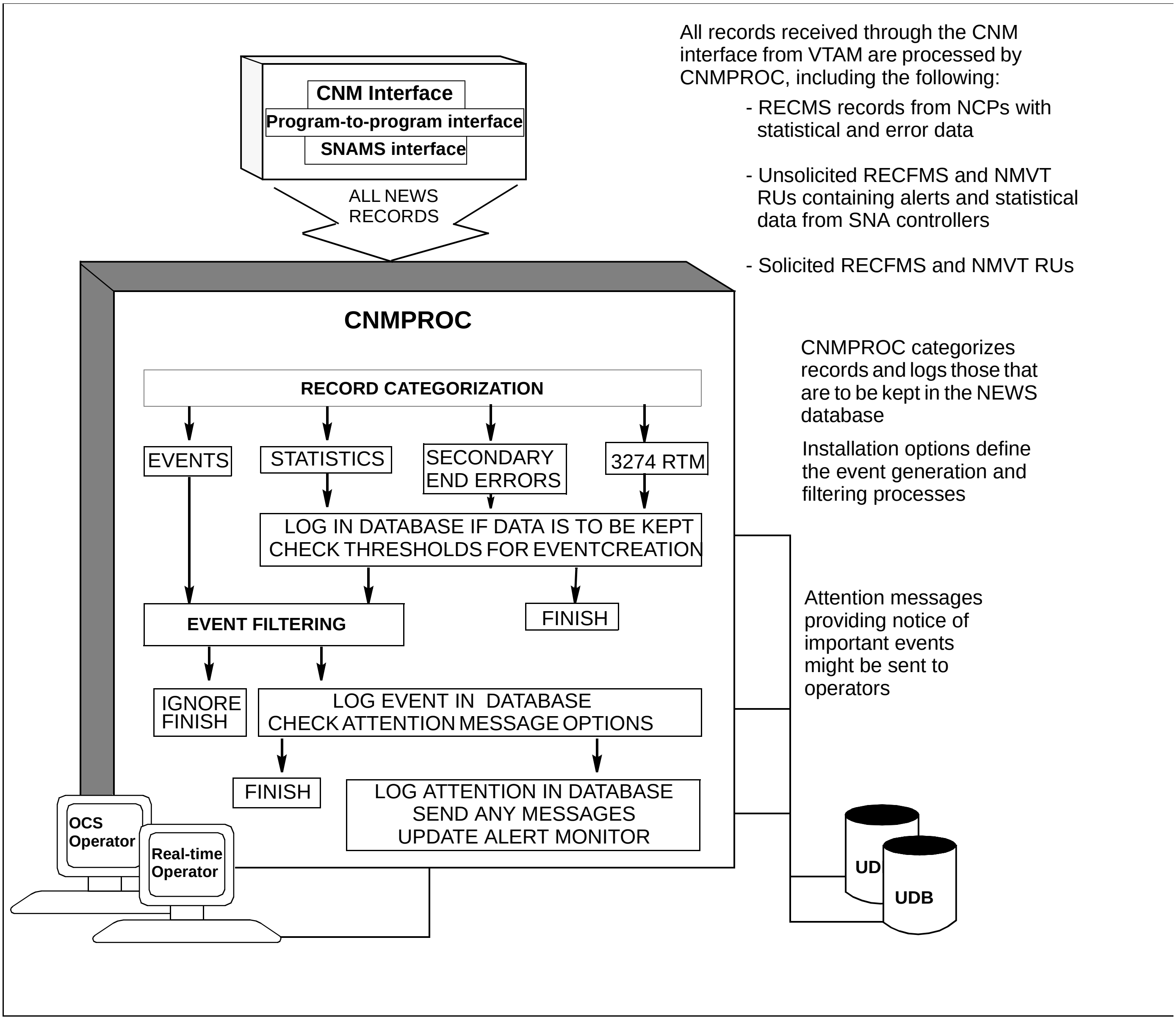
CNMPROC can be activated to process:

- All records that arrive unsolicited from VTAM across the CNM interface
- Records from APPN nodes sent to the ALERT-NETOP application using the Management Services implementation of MS Transport
- Solicited records delivered to it by users
- Alerts created by the &CNMALERT verb

Messages generated by CNMPROC are sent to OCS users who have monitor capability, and have a prefix of C to identify their origin.

Figure 2-1 shows how CNMPROC processes data arriving from the SNA network. See Appendix B, *About the CNM Interface*.

Figure 2-1. . How CNMPROC Processes Data from the SNA Network



All records received through the CNM interface from VTAM are processed by CNMPROC, including the following:

- RECMS records from NCPs with statistical and error data
- Unsolicited RECFMS and NMVT RUs containing alerts and statistical data from SNA controllers
- Solicited RECFMS and NMVT RUs

CNMPROC categorizes records and logs those that are to be kept in the NEWS database

Installation options define the event generation and filtering processes

Attention messages providing notice of important events might be sent to operators

NEWS Facilities

NEWS comprises a number of distinct facilities that combine to process records received from the local host.

NCL Verbs and Procedures

NEWS provides a variety of NCL procedures, as well as a number of NCL verbs and system variables, to perform different functions.

A summary of NEWS NCL verbs and system variables is given below:

&CNMALERT
Sends a CNM alert to a local or remote NEWS system for processing.

&CNMCLEAR
Clears any outstanding Response Units (RUs) which have been solicited by an &CNMSEND statement and not processed by an &CNMREAD statement.

&CNMCONT
Used within CNMPROC to send the current CNM record across specified ISR links.

&CNMDEL
Used within CNMPROC to delete the current CNM record or stop the current CNM record from being sent across specified ISR links.

&CNMPARSE
Produces tokenized data from the $CNM mapped MDO used by NEWS.

&CNMREAD
Makes the next CNM record received from VTAM available to CNMPROC, or the next outstanding RU available to a user procedure that has solicited data using an &CNMSEND statement.

&CNMSEND
Sends an RU across the CNM interface.

&CNMVECTR
Vectorize a CNM record which was previously segmented.

&NEWSAUTH
Indicates whether the user ID of the user invoking a procedure is authorized for NEWS (system variable).

&NEWSRSET
Indicates whether the user ID of the user invoking a procedure is authorized to delete records from the NEWS database (system variable).

&SNAMS
Provides the SNA Management Services interface which enables NCL procedures to participate as management applications within an APPN network.

For detailed information on NEWS NCL verbs and system variables, see the *Network Control Language Reference* manual.

Unattended Solicitation

NEWS supplies NCL procedures to solicit various types of network data, including RTM, VPD, EC level data, FCS, and LPDA2 data.

For information about the types of data solicited, see Appendix D, *NEWS Device Solicitation Procedures*.

NEWS Commands

A summary of NEWS commands is given below:

CNM
Starts and stops the VTAM CNM interface.

CNMTRACE
Defines a trace of records that come across the CNM interface. By default, all trace data is recorded.

DEFALIAS
Defines an alias entry for the Alias Name Translation Facility of NEWS.

DELALIAS
Deletes an alias entry used by the Alias Name Translation Facility of NEWS.

REPALIAS
Replaces an alias name entry used by the Alias Name Translation Facility of NEWS.

REQMS
Sends data across the CNM interface.

SHOW CNMTRACE
Displays active CNM trace requests.

SHOW DEFALIAS

Displays one or more DEFALIAS entries used by the Alias Name Translation Facility of NEWS.

SHOW SNAMS

Displays a list of all applications registered with SNA Management Services.

SYSMON

Logs the user on to the System Monitor residing in a 3600 or 4700 controller, and sends data to the Monitor.

SYSPARMS

Initializes or modifies system parameter values.

XLATE

Performs name translation testing through the Alias Name Translation Facility of NEWS.

For further information about these commands see the manual, *Management Services Command Reference*.

Alias Name Translation Facility

NEWS provides VTAM alias name translation services for those levels of VTAM that require this function. VTAM requests alias name translation from NetMaster for SNA if the CNM Routing table entry for the NetMaster for SNA CNM ACB contains the translate-inquiry RU. When establishing cross-domain or cross-network sessions, VTAM can request the translation of LU names, COS names, and LOGMODE names.

Generic name definitions allow ranges of names to be translated by NEWS from a single translation definition. You display and maintain translation definitions by using NEWS commands—see the section, *Maintaining Resource Alias Names*, on page 15-26.

Reviewing and Reporting on Data

After data has been filtered and recorded in the NEWS database, you can review it by using the NEWS menus and full-screen panels, or have it exported to a dataset for analysis by external systems.

If you want to analyze or report on NEWS data via an external system, you need to do one of the following:

- Use the supplied user exits to archive all required records to a sequential dataset.
- Generate SMF records by activating SMF recording (see the section, *Maintaining the NEWS Database*, on page 8-2).

If required, you can enable the generation of type 37 SMF records for all event, attention, and statistics records that pass NEWS filtering.

NEWS also provides predefined reports. For information about these reports, see the *Unicenter NetMaster Network Management for SNA User's Guide.*

3

How NTS Works

This chapter presents an overview of how the NTS component of NetMaster for SNA works, and introduces concepts that you need to understand to gain the maximum benefit from using this component.

This chapter discusses the following topics:

- Data Available to NTS
- How NTS Obtains Data
- Network Definitions and Names Used by NTS
- System Resource Utilization
- Session Start Processing
- Output Processing
- System Event Generation
- NTS Database
- MAI Support
- Use of ISR Links

Data Available to NTS

The primary objects that concern NTS are network addressable units, also called resources, and sessions between these resources. The resources can be in the same domain, in different domains, or even in different networks.

The data types available to NTS are:

- Session awareness (SAW) data, which consists of session start and session end notifications from VTAM
- Session data that contains information about the performance and status of a session, including:
  - Session trace data
  - Response time data
  - Route configuration data

Session Awareness (SAW) Data

To NTS, the local VTAM supplies SAW data relating to sessions that the local SSCP maintains. This includes data about SSCP-SSCP, SSCP-PU, SSCP-LU, LU-LU and CP-CP sessions.

If linked to other NTS systems, NTS can also receive SAW data from VTAMs in remote domains.

Composition

SAW data consists of the following data:

- *Session identification data*, including:
  - Session partner names (or aliases, if applicable) and addresses
  - PCIDs
  - Session start and end time
  - Session type and class (such as: same domain, cross domain)
- *Session connectivity data*, including:
  - Explicit Route (ER)
  - Virtual Route (VR) and Transmission Priority (TP) that the session is using
  - Logmode and Class of Service (CoS) table entries the session is using

- *Session hierarchy data*, including:
  - Controlling PU name
  - Link name, and subarea PU name (where relevant) for each session partner resource
- *Session exception data*, including:
  - Sense codes describing any error conditions that occurred while the session was in progress

Session Trace Data

Through NTS, you can issue requests to VTAM to start tracing sessions that involve resources in the local VTAM domain. As a result, NTS receives trace data from VTAM and associates it with session records in storage. NTS can also solicit trace data collected by VTAMs in other, linked NTS systems.

Composition

Trace data consists of copies of Path Information Units (PIUs) that flow on traced sessions. PIUs are message units that comprise:

- A Transmission Header (TH)
- A Request/response Header (RH)
- Any Request/response Unit (RU)

In the case of session control RUs, the entire RU is included. Otherwise, for performance reasons, only the first 11 bytes are retained. For details of how to obtain extended trace information, see the STRACE command description in the *Management Services Command Reference* manual.

Response Time Monitoring (RTM) Data

RTM data is a measure of how long it takes for an operation to be transmitted between a display station and a host.

NTS obtains the response time information from the cluster controller at session end, then associates the response time obtained for a display station with the session record in storage.

Note

In order for RTM data to be available to NTS, the cluster controller must support host programming. See your control unit’s customization guide for information on setting this option.

The cluster controller sends both solicited and unsolicited data to NTS. This data originates from PUs that implement RTM (3x74s or equivalent) for their attached LUs.

NTS can also solicit RTM data collected by other NTS systems.

Composition

RTM data received from the cluster controller consists of:

- *Boundary values in seconds*: these boundaries demarcate *buckets* into which individual response times are counted; an overflow bucket is also provided.
- *Bucket counts*: these counts represent the total number of response times within the specified boundaries since the beginning of the session, or since the response times were last reset.

Route Configuration Data

NTS receives both solicited and unsolicited route configuration data from VTAM and other subarea nodes. You can dynamically request ER and VR configuration information from subarea nodes visible to NTS.

Composition

Route configuration information includes:

- Source, destination, and adjacent subarea numbers
- Source, destination, and adjacent control point names if the session involves APPN
- The status of the ER, VR, and TP
- Session route information for the current APPN subnetwork if available

How NTS Obtains Data

NTS derives its data from these sources:

- Standard host access method interfaces (VTAM)
- Other NTS systems
- The Multiple Application Interface (MAI) component of SOLVE:Access

From VTAM Interfaces

VTAM is aware of all sessions that have at least one session partner defined in its domain. These sessions are presented to the local NTS system across a VTAM interface, resulting in NTS building up an image of the logical network activity within this domain.

VTAM interfaces to NTS include:

- A CNM interface, through which NTS requests are issued to VTAM, and RTM, VR, and ER information is collected
- A local VTAM interface, through which the following sessions are conducted:
  - An LU-LU session with VTAM for the collection of SAW data (and some route configuration data)
  - An LU-LU session with VTAM for the collection of session trace data

The local VTAM interface is called ISTPDCLU. To define the NTS side of the interface, complete the NTS User Exit Details page of the SAW parameter group in ICS. See the *Starting NetMaster for SNA for the First Time* chapter in the *Unicenter NetMaster Network Management for SNA Implementation Guide*.

For information about controlling NTS data collection, see Chapter 9, *Tailoring NTS*.

Through Inter-system Routing Between NTS Systems

NTS uses the Inter-System Routing (ISR) feature of Management Services to obtain further information about cross-domain and cross-network sessions.

VTAM is only aware of sessions that have at least one session partner defined in its domain. It is possible to centralize (or distribute) the monitoring of logical network activity by expanding the sources of data available to an NTS systems to include:

- SAW data collected by NTS systems in other domains
- Session trace, accounting, and RTM data collected by NTS systems in other domains

The other NTS systems may or may not be within the same SNA network.

Single Image Presentation

The user of an NTS system that is linked to other NTS systems is presented with a *single image* of:

- The sessions between resources throughout the network(s)
- The performance and problem determination data collected for these sessions

This single image is preserved in the NTS database and NTS SMF exit.

Complete knowledge of a session partner in another VTAM domain is available only if the domains are connected by an INMC link that supports Inter-System-Routing (ISR) processing. NTS uses the facilities of appropriately connected and configured ISR links between systems to exchange information. To define ISR links, complete the ISRIN and ISROUT parameter groups in ICS.

For information about completing the ISRIN and ISROUT parameter groups, see the section*, Defining Communications for NEWS and for NTS*, on page 15-14.

For information about ISR links, see the *Management Services Administrator Guide*.

Through MAI Sessions

MAI is a component of the SOLVE:Access product that allows a user to operate multiple sessions concurrently. See the section, *MAI Support*, on page 3-10 for further details.

MAI provides NTS with information about the logical relationship between the real half sessions that form the MAI virtual session. When the MAI/NTS interface is first activated, MAI provides NTS with this information for all currently existing MAI sessions. MAI then notifies NTS as new MAI sessions are started.

Network Definitions and Names Used by NTS

NTS does not require definitions of the network or VTAM environment in which it is executing. All such knowledge is derived by NTS through standard access method interfaces. NTS panels display the network ID for the host VTAM under which NTS runs.

Because NTS operates in SNA Network Interconnection (SNI) environments, all NTS resource names are qualified by the network name (that is, both the resource name and the network name are required to identify an SNA resource), and alias names are fully supported. The network in which NTS is executing is always the assumed default. Therefore, the use of network qualified names by NTS in a single network environment (or within the default network in an SNI environment) is totally transparent to the user.

System Resource Utilization

NTS defines buffer pools for allocating all resource records, session records, and trace data kept in virtual storage. No storage is allocated until NTS begins SAW processing. Both storage allocation and NTS processing are carefully managed to produce low system overheads and paging rates. In MVS/XA and above systems, all NTS buffers are located above the 16MB line.

It is possible to retain all session information and to trace all session activity on most installations, with little or no loss of performance. This does, of course, depend on the extent of system resource consumption during network operation.

Session Start Processing

As part of session start notification, this processing occurs:

1. VTAM passes to NTS the SSCP name or names that identify the domain or domains in which the participating resources reside.
2. If either resource does not reside in the local domain, NTS determines whether an ISR link to the NTS in the other domain exists.
3. If there is a suitably configured link, NTS is able to solicit session data from the other domain; if not, data for the session will be incomplete.
4. NTS then proceeds to classify the session.

Output Processing

Session records can be queued for output processing by NTS for any of these reasons:

- Session start notification—applies only to sessions for which accounting information has been requested
- Forced logging by an operator through an NTSMOD command
- Session awareness close processing
- Normal end-of-session processing

For sessions placed on the output queue, logging commences immediately after one of the following:

- Session start notification (accounting information is logged)
- Being force-logged by an operator
- Being closed by session awareness close processing

Session End Processing

All data associated with an active session is kept in storage until the session ends. When termination notification for a session is received from VTAM, NTS queues the session for output processing, which occurs as follows:

1. Firstly, NTS correlates all session data, such as session awareness data and any other statistical or problem-determination data, waiting for the correlation interval if necessary.
2. This final session data is then passed to a user exit, if one is active, and logged in the NTS database (unless the user exit suppresses logging).
3. When output processing is complete, the session data is purged from storage and is subsequently available (in the NTS database) as historical data only.

NTS User Exit Processing

If an exit is defined, all session records scheduled for logging (this is determined by the record type) are first passed to this exit. The exit can perform additional record processing, and can set a flag to indicate that the record be ignored by the subsequent NTS logging function.

For further information about the NTS User Exit and its functions, see Appendix E, *Implementing the NEWS User Exit*. The exit is also described fully in the documentation included with the sample exit, and within the macro DSECT named $NMSMF, which are provided with NTS. The sample exit provides the record layout for the NTS output data to be written to the system SMF repository.

System Event Generation

Event Distribution Services (EDS) is a component of Management Services that allows NCL procedures to *listen* for and generate *events.*

If requested by means of SYSPARMS NTSEVENT, EDS generates events on behalf of NTS at the following times:

Time	**Event Name**
At session start	$$NTS.SESSION.START
At session end	$$NTS.SESSION.END
On session failure	$$NTS.SESSION.FAIL
When RTM objectives are exceeded	$$NTS.RTM.OBJ.EXCEEDED

A data field containing all details relating to the session accompanies the event notification. Any SNA sense code supplied appears in the reference code field of the event notification.

NTS Database

All information logged to the NTS database is session-related and is stored under the session partner names. Together, the two network-qualified session partner names form a *session name pair* and each session logged in the database is termed a *session incidence*. For each session name pair, there exists a master and a cross-reference record, both of which are created when the first session for the name pair is logged to the database. (For an example of how you can limit the number of session name pairs stored in the NTS database, see the section, *Defining Session Classes*, on page 9-3.)

Session Keep Counts and Database Slots

The session incidence count for any given session name pair is restricted by the *session keep count*. The default session keep count is 10, but this can be modified—see Chapter 9, *Tailoring NTS*, in this manual. This value is stored as part of the master record when the first session incidence for a session name pair is logged.

Each session incidence is allocated a single *slot* in the database. When a new session is due to be logged, the master record is checked to determine whether the number of slots used for the session name pair has yet reached the session keep count. If it has, then the oldest session incidence data is overwritten; otherwise, a new database slot is allocated.

The advantage of database slots is that the key used to access session incidence data can be reused, which means that database maintenance is minimized. For example, if the database currently contains as much data as it will ever be required to hold, then it can be used for session logging indefinitely without requiring reorganization. It takes some time before the database reaches such an ideal state, however.

An example of an implementation strategy and details of the space requirements for the NTS database are given in Appendix I, *NTS Storage Estimates*.

Connecting to the NTS Database

By default, each time session awareness processing is started, NTS attempts to connect to the NTS database and prepares for logging. NTS searches internally to find a file ID of NTSLOG, in the same way that an NCL procedure searches for a file to connect to. This enables the installation to determine both the dataset name and DD name of the NTS database, and allows the database to be opened via any of the options available on the UDBCTL command. (The UDBCTL command is a Management Services (MS) command used to open VSAM files and to optionally allocate an internal file ID. See the *Management Services Command Reference* for a complete description of these commands.)

If an error occurs in the NTS database during output processing, the NTSLOG file ID is released. NTS continues to function normally without a database, apart from the fact that it cannot perform database logging until you allocate and open a new database. (See the section on connecting and disconnecting the NTS database in Chapter 11, *Maintaining NTS*.)

MAI Support

The Multiple Application Interface (MAI) component of SOLVE:Access enables you to operate multiple sessions simultaneously For more information about MAI, see the *SOLVE:Access User's Guide* or the *SOLVE:Access Implementation and Administration Guide*.

What Is an MAI Session?

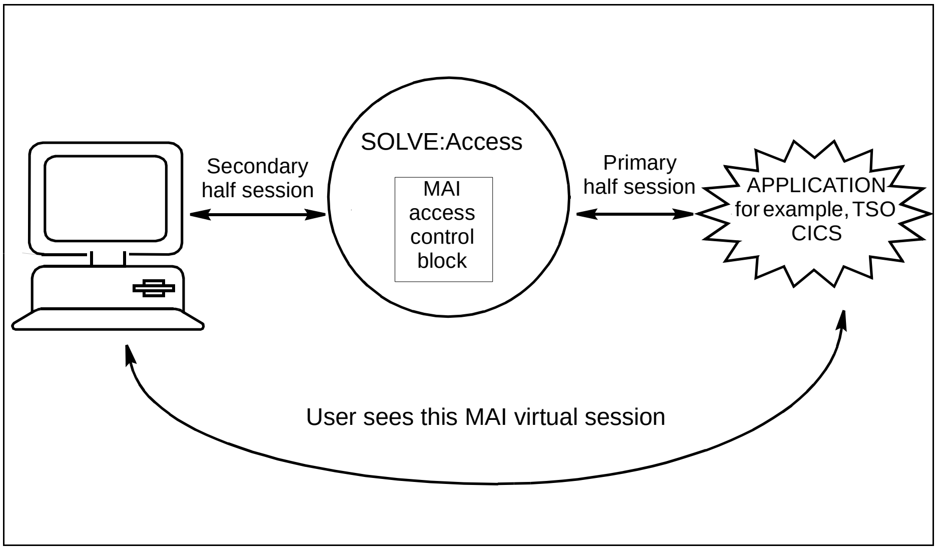
An *active* MAI session consists of two real SNA sessions. MAI relates these sessions by transferring data received for one session across to the other. The result is that, to the user, the endpoints of two distinct sessions appear to be in session with one another.

For the purposes of identification, the sessions related by MAI are termed the primary and secondary *half sessions* of an MAI *virtual session.* These half sessions, which are transparent to the MAI user, comprise the following elements:

- The primary half session has an application as its primary session partner and an MAI ACB (an ACB defined to SOLVE:Access for the use of MAI) as its secondary session partner.
- The secondary half session has SOLVE:Access as its primary partner and, in most cases, a terminal as its secondary partner.

This process is illustrated in Figure 3-1.

Figure 3-1. MAI Session



VTAM presents NTS with SAW data for each of the MAI half sessions, but is unaware that they are logically related. MAI, via the NTS/MAI interface, advises NTS that the half sessions are logically related.

This relationship is presented on a special MAI Session Configuration panel. (You can display a Session Configuration panel for each of the half sessions.) See the *NetMaster for SNA User's Guide* for more information about the NTS Session Configuration panel.

The MAI/NTS Interface

When active, NTS is aware of all real sessions that have at least one partner in its domain. Using the NTS/MAI interface, MAI provides NTS with information about the logical relationship between the real half sessions that form the MAI virtual session or sessions.

From this, NTS builds information from the two half sessions into a single virtual session. This virtual session can be listed, selected, and displayed in the same manner as real sessions within NTS.

No RTM data is collected or available for MAI sessions.

Logging MAI Sessions to the NTS Database

When an MAI session is logged to the NTS database, NTS checks if any trace or accounting data, or both, is flagged as available for the MAI session. If this is the case, NTS logs the trace or accounting data (or both) collected for the primary half session with the MAI session incidence record. (This avoids the need to log the primary half session to the database if the MAI session is the preferred record of the session incidence.)

Logging MAI Sessions to the NTS User Exit

The user ID of the user who started the MAI session is provided by the MAI/NTS interface. This data is passed to the NTS user exit in an additional field added to the session configuration section of the type 39 SMF record (see Appendix G, *NTS SMF Record Formats*). This field contains nulls for non-MAI sessions.

Use of ISR Links

A large installation can have many systems active in many domains, and on a number of hosts. These systems can be linked by ISR links.

ISR links can be used by NTS in two ways:

- For session configuration data sharing
- By NTS Single Image

Session Configuration Data Sharing

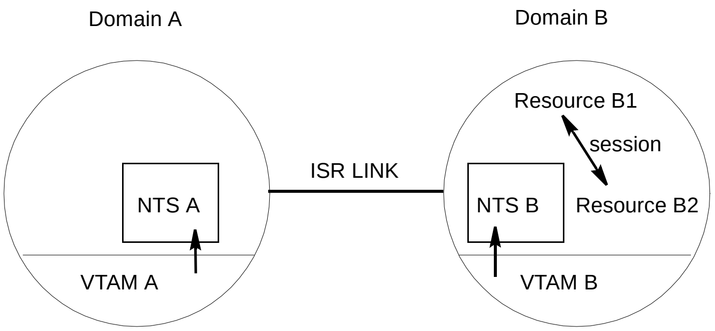
In a multi-host environment, the session configuration data available to NTS for a cross-domain session is incomplete, because only the information relating to the session partner owned by the local VTAM system is visible to NTS. However, if another NTS system is running on the remote VTAM to which the other session partner is defined, it will have captured the missing data. Under such circumstances, the respective NTS systems are in a position to exchange data, so that both systems can complete their records.

The exchange of session configuration data occurs automatically if there is an ISR link between the two systems, as shown in Figure 3-2.

Note

Links should be established before session awareness processing is activated, to ensure the collection of complete session configuration data.

Figure 3-2. Transfer of Session Data via an ISR Link



VTAM in domain B supplies NTS B with data relating to the session between resources B1 and B2, which in turn is forwarded to NTS A.

Arrowheads indicate flow of data.

NTS Single Image

NTS-SI uses ISR to share session awareness and session configuration data between NTS systems, when the remote NTS systems have available session data that would not normally be available to the local NTS system. If you configure your ISR links correctly, this session information can be shared amongst NTS systems (see *How NTS-SI Works*, on page C-14).

4

How NCS Works

Network Control Services (NCS) provides summary displays of resource types, and graphic displays of individual resources and their subordinate nodes.

This chapter discusses the following topic:

- How NCS Works

How NCS Works

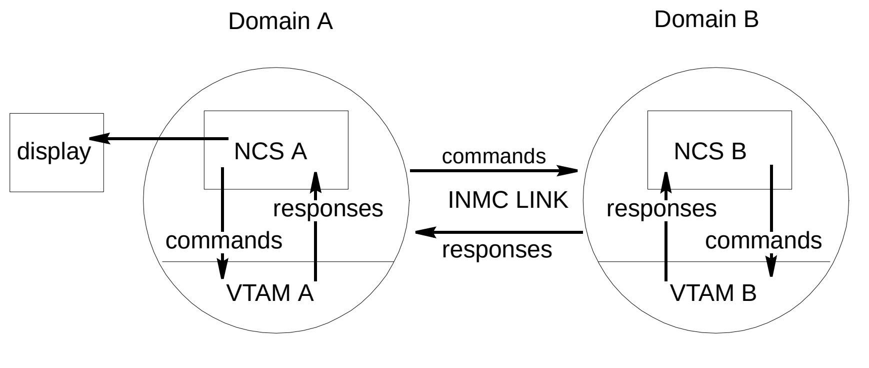
NCS functions by issuing VTAM Display commands and interpreting the responses received.

It also collaborates with other, linked NetMaster for SNA systems, by using the Inter-Management Services Connection (INMC) facility. VTAM Display commands can be executed on other NetMaster for SNA systems in other VTAM domains for processing, and responses can be returned to the original NCS for display, as shown in Figure 4-1.

Note

This can occur only if the user has access to the other NCS, and the user IDs in both systems are the same.

Figure 4-1. Transfer of NCS Data Across an INMC Link



NCS in domain A sends VTAM Display command requests to NCS in domain B, which returns responses. These responses can be displayed by NCS A.

Arrowheads indicate flow of data.

Access to Configuration Data

To be able to list configuration data, NCS needs to be integrated with a configuration management database. See Chapter 12, *Tailoring NCS*.

5

How NCPView Works

NCPView monitors IBM 3720, 3725, 3745, and 3746-900 communications processors that are running NCP version 4, 5, 6, or 7.

Note

Although NCPView can obtain information about a 3746-900 communications processor, it cannot communicate directly with this type of resource. NCPView obtains what information it can about a 3746-900 communications processor from the associated 3745 processor.

This chapter discusses the following topic:

- How NCPView Works

How NCPView Works

NCPView obtains NCP data via the VTAM secondary program operator (SPO) interface, by using VTAM display commands.

NCPView identifies NCPs at initialization by issuing a D RSCLIST,IDTYPE=PUT45 command.

NCPView obtains its information about the NCPs via the standard VTAM command—D NET,NCPSTOR,ADDR=*xx*.

Alternatively, NCPView can obtain its information from an NCP unformatted dump. This is achieved by NCPView reading a section of storage in the unformatted dump instead of issuing a D NET,NCPSTOR,ADDR=*xx* command.

See Chapter 13, *Tailoring and Controlling NCPView*.

NCPView and Connected Domains

You can view all the NCPs in your enterprise from one domain, provided that you are operating in a multisystem environment supported by Automation Services.

For further details, see the chapter, *Administering a Multisystem Environment*, in the *Automation Services Administrator Guide*.

Part II

Administering NetMaster for SNA

6

Administering NetMaster for SNA

This chapter describes the following topics:

- Performing Administrative Tasks
- Implementing Features
- Customizing Facilities
- Tailoring the NetMaster for SNA Startup Procedure
- About Security
- Defining User Exits

Performing Administrative Tasks

To use the NetMaster for SNA facilities that have been authorized for your use, you can perform various administrative tasks to implement and customize the facilities to suit your installation.

Implementing Features

The tasks required to implement various features are described in the following sections. Many of these tasks are performed using ICS parameter groups, as shown in Table 6-1.

Table 6-1. Implementing NetMaster for SNA Features

To implement...	**Use parameter group...**	**See section...**
The Network Services Control file	NSCNTL	*Implementing the Network Services Control File (NSCNTL)*, on page 6-3
NEWS databases	NEWS	*Implementing NEWS Databases*, on page 6-3
The NTS log database	NTS	*Implementing the NTS Log Database*, on page 6-4
Your initialization procedure	SNAINIT	*Identifying Your Initialization Procedure (SNAINIT)*, on page 6-5
NSCNTL cache options	NSCNTLCACHE	*Implementing NSCNTL Cache Options*, on page 6-5
NEWS database logging options	NEWSDBOPTS	*Implementing NEWS Database Logging Options (NEWSDBOPTS)*, on page 6-6
Event filters	CNMFILTERS	*Implementing Event Filters (CNMFILTERS)*, on page 6-7
Performance objectives for event recording	CNMPERFOBJ	*Implementing Performance Objectives for Event Recording (CNMPERFOBJ)*, on page 6-8
SMF event recording options	SMFT37	*Implementing SMF Event Recording Options (SMFT37)*, on page 6-8

Table 6-1. Implementing NetMaster for SNA Features

To implement...	Use parameter group...	See section...
The PPI receiver	PPINETVALRT	*Implementing the PPI Receiver (PPINETVALRT)*, on page 6-9
Device support	DEVICESUPP	*Implementing Device Support Diagnostics (DEVICESUPP)*, on page 6-9

Implementing the Network Services Control File (NSCNTL)

You need to define a Network Services (NSCNTL) database for your product region. To do this:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the NSCNTL parameter group. The NSCNTL - NSCNTL File Specifications panel is displayed.

Step 3. Enter the NSCNTL File ID. This specifies the file name of your Network Services (NSCNTL) database, which is a required database.

Step 4. For information on the remaining fields, press F1 (Help).

Step 5. Press F6 (Action) to action your entries.

Step 6. Press F3 (File) to save your settings.

Implementing NEWS Databases

You can define the following NEWS databases for your product region:

- Network Error (NEWSFILE) database
- Network Error Backup (NEWSBKP) database.

Note

If you do not define the NEWSFILE database, no CNM events can be saved. If you do not define the NEWSBKP database, you cannot perform an online reorganization of the NEWSFILE database.

To define the NEWSFILE and NEWSBKP databases:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the NEWS parameter group. The NEWS - NEWS File Specifications panel is displayed.

Step 3. Enter the NEWS Database File ID. This specifies the file name of your Network Error database. If you do not enter a value here, no Network Error database is allocated.

Step 4. Complete the remaining fields on the panel. For information on the fields, press F1 (Help).

Step 5. Press F8. The second panel for this parameter group is displayed.

Step 6. Enter the NEWSBKP Database File ID. This specifies the file name of your Network Error Backup database. If you do not enter a value here, no Network Error database is allocated.

Step 7. Complete the remaining fields on the panel. For information on the fields, press F1 (Help).

Step 8. Press F6 (Action) to action your entries.

Step 9. Press F3 (File) to save your settings.

Implementing the NTS Log Database

You can define a Network Tracking Log (NTSLOG) database for your product region.

Note

If you do not define the NTSLOG database, no session awareness or session trace data can be saved.

To define the NTSLOG database:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the NTS parameter group. The NTS - NTSLOG File Specifications panel is displayed.

Step 3. Enter the NTSLOG Database File ID. This specifies the file name of your Network Tracking Log database. If you do not enter a value here, no NTSLOG database is allocated.

Step 4. Complete the remaining fields on the panel. For information on the fields, press F1 (Help).

Step 5. Press F6 (Action) to action your entries.

Step 6. Press F3 (File) to save your settings.

Identifying Your Initialization Procedure (SNAINIT)

A default initialization procedure, $NSINIT, is provided with NetMaster for SNA. If you make a copy of this procedure and customize it, you need to identify your customized initialization procedure to NetMaster for SNA.

To identify a customized initialization procedure:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the SNAINIT parameter group. The SNAINIT - NetMaster for SNA Init Process panel is displayed.

Step 3. If you have copied and customized the default initialization process, $NSINIT, enter your process name under NetMaster for SNA Initialization Details.

Step 4. Press F6 (Action) to action your entries.

Step 5. Press F3 (File) to save your settings.

Implementing NSCNTL Cache Options

By setting an optimum cache size for the Network Services Control File, you can improve the performance of NEWS processing by eliminating as much VSAM activity as possible.

The Control File is a database that controls all NEWS CNM record processing. The Control File contains codes and messages that are used to control NEWS functions. It is front-ended by a cache that holds the most recently retrieved records from the Control File. Once full, the cache discards an infrequently referenced record when a new record is added.

Note

The cache is an in-storage vartable. Actioning this parameter group results in the vartable being deleted and redefined.

To implement the NSCNTL cache options:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the NSCNTLCACHE parameter group. The NSCNTLCACHE - NSCNTL Cache Size panel is displayed.

Step 3. Enter a value in the Maximum Number of Records Cached field, or leave the default value.

Step 4. Press F6 (Action) to action your entries.

Step 5. Press F3 (File) to save your settings.

Implementing NEWS Database Logging Options (NEWSDBOPTS)

You can control the performance of record logging to the NEWS database by implementing the NEWS Database Recording Options parameter group. This parameter group controls how many records are stored on the NEWS database for each resource name, per category.

To implement these options:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the NEWSDBOPTS parameter group. The NEWSDBOPTS - NEWS Database Recording Options panel is displayed.

Step 3. For each category shown, enter the maximum number of records to be captured, or leave the default value. For further information, press F1 (Help).

Step 4. Press F6 (Action) to action your entries.

Step 5. Press F3 (File) to save your settings.

Examples

To receive a warning message every time a device information record is discarded, type the value **1** in the Device Information field.

To receive no warnings after the initial notification that logging has been suspended, type **0** in the Device Information field.

Implementing Event Filters (CNMFILTERS)

You can control how NetMaster for SNA processes network events by defining which events are to be recorded, how they are to be processed, and what severity alert NetMaster for SNA is to generate. You do this by implementing the Event Recording Filters parameter group.

The use of resource masks restricts the processing of event records to only a subset of the network. The Event Recording Filters parameter group allows you to set resource masks for further filtering of records within the selected event type. You can specify any combination of include or exclude mask type for an event type.

To implement the Event Recording Filters parameter group:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the CNMFILTERS parameter group. The CNMFILTERS - Event Recording Filters panel is displayed. This panel has ten pages (each with two types of event filter) that you can scroll through to define filters for each event category. For a list of event types, see the section, *Events*, on page 2-6.

Step 3. For each event category, enter the Processing Option and Alert Severity, if you want to change the default values.

Step 4. For each event category for which you want further filtering, define an Include Mask or an Exclude Mask. For further information, press F1 (Help).

Step 5. Press F6 (Action) to action your entries.

Step 6. Press F3 (File) to save your settings.

Examples

To prevent an event generating an alert that is displayed on the Alert Monitor, specify an Alert Severity of **0**.

To write an event to the Events Category, specify a Processing Option of **E**.

Implementing Performance Objectives for Event Recording (CNMPERFOBJ)

The Performance Objectives parameter group allows you to set threshold values for certain network statistics that cause a performance event to be created when the values are reached or exceeded.

The network statistics used for performance objectives are:

- 3x74 RTM Data
- RECMS Statistics
- FCS RECFMS 04 Data

To define performance objectives:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the CNMPERFOBJ parameter group. The CNMPERFOBJ - Performance Objectives panel is displayed. This panel has three pages that you can scroll through to define performance objectives for each type of event.

Step 3. For each type of event, enter the threshold values for a performance event to be created. For further information, press F1 (Help).

Step 4. Press F6 (Action) to action your entries.

Step 5. Press F3 (File) to save your settings.

Implementing SMF Event Recording Options (SMFT37)

By setting SMF recording options, you can control the generation of SMF records from NEWS events and statistics.

When turned on, these options cause CNMPROC to write the specified NEWS records to the SMF file in type 37 format.

To set these options:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the SMFT37 parameter group. The SMFT37 - SMF Type 37 Recording Options panel is displayed.

Step 3. Enter a value in the Events and Attentions field to specify the types of NEWS event records for which you want SMF records to be written. For further information, press F1 (Help).

Step 4. Enter a value in the Statistics field to specify whether you want SMF records to be written for statistics.

Step 5. Press F6 (Action) to action your entries.

Step 6. Press F3 (File) to save your settings.

Implementing the PPI Receiver (PPINETVALRT)

The PPI receiver processes external events that are queued to the NETVALRT PPI resource by other tasks. To define details for starting and stopping the PPI receiver:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the PPINETVALRT parameter group. The PPINETVALRT - NETVALRT PPI Receiver Process panel is displayed.

Step 3. Specify **YES** or **NO** in the Initially Active? field. This field indicates whether to start the PPI receiver during initialization.

Step 4. Specify **YES** or **NO** in the Currently Active? field. This field indicates whether to start or stop the PPI receiver now.

Step 5. Press F6 (Action) to action your entries.

Step 6. Press F3 (File) to save your settings.

Implementing Device Support Diagnostics (DEVICESUPP)

News device support allows you to diagnose problems with SNA controller devices. The DEVICESUPP parameter group allows you to specify which options are to be displayed on your SNA : Device Support Diagnostic Menu. To do this:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the DEVICESUPP parameter group. The DEVICESUPP - Device Support Diagnostics Menu panel is displayed.

Step 3. For each option shown, enter YES or NO to control whether the option is displayed on the Device Support Diagnostics Menu. For further information, press F1 (Help).

Step 4. Press F6 (Action) to action your entries.

Step 5. Press F3 (File) to save your settings.

Customizing Facilities

The tasks required to customize facilities for various features are described in the following sections.

Customizing NEWS Facilities

To customize NEWS facilities in your NetMaster for SNA region, you can update the following parameter groups that you reviewed when implementing your region:

- NEWS File Specifications
- NEWS Database Logging Options

See the *Unicenter NetMaster Network Management for SNA Implementation Guide*.

You can also do the following:

- Review your NEWS parameters to suit your installation. For detailed steps about tailoring NEWS parameters and operational characteristics, see Chapter 7, *Tailoring NEWS*.
- Manage your network focal points and entry points by using the NEWS : Control Functions Menu. For details about managing focal points and entry points, see Chapter 8, *Maintaining the NEWS Database*.

Customizing NTS Facilities

To customize NTS functions to suit your installation, use the SYSPARMS and DEFCLASS commands. These commands are normally included in the NetMaster for SNA initialization procedure, $NSINIT.

For details about the $NSINIT procedure review, see the section, *Tailoring the NetMaster for SNA Startup Procedure*, on page 6-11.

Customizing NCPView Facilities

To customize NCPView facilities to suit your installation, you can:

- Tailor the NCPView NCL exit.
- Define additional NCPs to be monitored.

For further details, see Chapter 13, *Tailoring and Controlling NCPView*.

Customizing NCS Facilities

To customize NCS facilities in your NetMaster for SNA region, you can:

- Limit the size of display lists and individual node displays
- Integrate NCS with a configuration management database

For further details, see Chapter 12, *Tailoring NCS*.

Tailoring the NetMaster for SNA Startup Procedure

The $NSINIT procedure is executed during NetMaster for SNA initialization and is designed for actions specific to NetMaster for SNA .

You must review the $NSINIT procedure for the following purposes:

- To enable NetView operator command emulation, if required.
- To set up NTS.

Note

$NSINIT is the default procedure. If you copy this procedure and customize it, you need to identify your customized procedure in the SNAINIT parameter group in ICS (see *Identifying Your Initialization Procedure (SNAINIT)*, on page 6-5).

Enabling NetView Operator Command Emulation

If you have NetView at your installation, you can use the same NetView operator commands in NetMaster for SNA. To enable this facility, perform these steps:

Step 1. Add the following statement to the $NSINIT member.

```
EXEC $VWCALL OPT=INIT
```

Alternatively, if this statement already exists, remove the comment symbols (-*) beside the statement.

Step 2. Review the command emulation table entries contained in the CAS table with the name EQUATES and application ID of $VW. For information on reviewing and modifying table entries, see the section, *Modifying Table Entries*, on page 14-2.

Step 3. Action the SNAINIT parameter group in ICS for the changes to take effect. See *Identifying Your Initialization Procedure (SNAINIT)*, on page 6-5.

Setting Up NTS

In the $NSINIT procedure, review the following:

- SYSPARMS parameters to control NTS functions
- DEFCLASS commands to control NTS data collection

These are administrative tasks that you can perform either now or after startup. For details about the SYSPARMS parameters for NTS, and how to set up NTS data collection, see the following:

- Chapter 9, *Tailoring NTS*, and Chapter 10, *Collecting NTS Data*, in this manual
- The appendix titled *SYSPARMS Operands* in the *Management Services Administrator Guide*

About Security

Access to a product region is controlled by the User ID Access Maintenance Subsystem (UAMS).

NetMaster for SNA supplies five sample group definitions that are generated during installation. These groups and their characteristics are described below:

- $RMADMIN—administrator—this group of users has access to all NetMaster administrative functions, such as adding users. An administrator has access to all the menu options and is authorized to delete database records.
- $RMBUSER—background user—this group of users has region or engine component authorization.
- $RMMON—monitor—this group of users has access to a restricted subset of NetMaster functions. A monitor user does not have access to all the menu options and can browse but not update or delete database records.
- $RMNOPER—network operator—this group of users has similar access to NetMaster functions as an operator. Network operators can manage network operations but are not authorized to manage system operations.

- $RMOPER—operator—this group of users has access to a restricted subset of NetMaster functions. An operator does not have access to all the menu options and is not authorized to delete database records.

> **Note**
>
> Do not modify the supplied $RMBUSER group definition, because this could impede the operation of the NetMaster product.

Security Considerations for Existing Users

If you are using a pre-existing UAMS database, perform the following tasks to ensure that users are properly authorized to operate in the region:

- Ensure that the group definitions are authorized for NetMaster for SNA.
- Ensure that the background users are defined by using the $RMBUSER group definition.

Checking Existing User Group Definitions

Ensure that the group definitions are authorized for NetMaster for SNA as follows:

Step 1. Enter the /UAMS shortcut to access the UAMS : Primary Menu.

Step 2. Type **L** at the ===> prompt and **$RM** in the User field. The group definitions are listed.

Step 3. Update each of the $RMADMIN, $RMBUSER, $RMMON, $RMNOPER, and $RMOPER definitions to ensure that the following fields are specified correctly:

Panel	Field	Value
3rd	Network Management field on the Access Authorities panel	Y
8th	NEWS Access field on the Network Management Details panel	Y
8th	Reset Authority field on the Network Management Details panel	N for $RMMON; Y for others
8th	NTS Access field on the Network Management Details panel	Y
8th	NCS Access field on the Network Management Details panel	Y

8th	NCPView Authority field on the Network Management Details panel	1

Customizing Existing Background User Definitions

The $RMBUSER group definition for background regions is defined when a product region starts the first time. The following UAMS background user definitions (where *xxxx* is the domain ID) are also generated and linked to the UAMS group:

User ID	**Description**
*xxxx*AOMP	AOM procedure
*xxxx*BLOG	Logger
*xxxx*BMON	Monitor
*xxxx*BSVR	Server
*xxxx*BSYS	System
*xxxx*CNMP	CNM procedure
*xxxx*LOGP	Log procedure

Note

The domain ID of the region is specified in the RUNSYSIN member during setup.

Updating Background User Definitions

If you set up your region by using a pre-existing UAMS database in which the background users are already defined for your region, those background user definitions are not replaced. To enable the new region to work correctly, you must update those background user definitions by associating the definitions to the $RMBUSER Group ID. You can do this by completing the following steps:

Step 1. Enter the **/UAMS** shortcut. The UAMS : Primary Menu is displayed.

Step 2. Enter **L** at the ===> prompt and ***xxxx*** in the User field. A list of the background user definitions is displayed.

Step 3. Update the background user ID by entering $RMBUSER in the Group ID field.

Step 4. Press F3 (File) to file the change.

Step 5. Repeat steps 3 and 4 for each of the background user IDs.

Step 6. After you finish updating the user definitions, enter **=CMD** at the ===> prompt to access the NetMaster : Command Entry panel.

Step 7. Enter the following command for each of the background users to invoke the changes.

```
SUBMIT USER=background-user-id SIGNON
```

The following example invokes the changes for the background logger:

```
SUBMIT USER=xxxxBLOG SIGNON
```

Defining User Exits

If you have NEWS or NTS user exits, then define them to NetMaster for SNA by using the CNM and SAW parameter groups in ICS. See the chapter titled *Starting NetMaster for SNA for the First Time* in the *Unicenter NetMaster Network Management for SNA Implementation Guide*.

For details about how to implement the NEWS user exit, see Appendix E, *Implementing the NEWS User Exit*, in this manual.

For details about how to implement the NTS user exit, see Appendix F, *Implementing the NTS User Exit*, in this manual.

7

Tailoring NEWS

NEWS can be operated and used immediately following the installation of NetMaster for SNA. However, you can at any time change certain parameters to tailor its operational characteristics.

This chapter discusses the ways in which you tailor the use and operation of NEWS to the requirements of your installation. The information is presented as a series of steps to be performed. Each step requires Management Services System Support functions to be performed online.

This chapter describes the following topics:

- Reviewing Parameters to Send and Receive CNM Data
- Reviewing NCP Parameters and Operations
- Customizing Device Configuration

Reviewing Parameters to Send and Receive CNM Data

To send and receive CNM data, you need to set fields on the first page of the CNM parameter group.

In the CNM parameter group, specify the required CNM ACB name. The CNM ACB name is the name of the ACB used to send and receive CNM requests and responses, and optionally, used to receive unsolicited CNM data.

Note

If the ACB specified is unable to receive unsolicited CNM data, then the name of the NetMaster for SNA or NetView region where it resides must be specified in the ISR Link Name field on the ISRIN Initialization Parameters panel.

See the section, *Defining Communications for NEWS and for NTS*, on page 15-14.

Tailoring NEWS Database Options

The NEWS database options allow you to control how many records are stored on the NEWS database for each resource name, per category.

To implement the NEWS database options, use the NEWSDBOPTS parameter group in ICS. See the section, *Implementing NEWS Database Logging Options (NEWSDBOPTS)*, on page 6-6.

Reviewing NCP Parameters and Operations

The NCP generates statistics records whenever certain internal counters overflow. Generally speaking, the counters for SNA devices overflow fairly frequently because the transmission counters include poll-type transmissions, and so the arrival rate of statistics for those devices is normally high. The counters for non-SNA devices, however, include only data transmissions, and by default wrap only after 65535 transmissions or 255 temporary errors. The arrival rate for statistics records for these devices can be quite low.

To adjust the statistics arrival rate for both SNA and non-SNA devices, specify the SRT NCP generation parameter. You might specify it on a PU macro for an SNA controller, or a TERMINAL macro for a non-SNA terminal. It is suggested that the parameter be utilized, particularly for non-SNA devices, so that statistics can be kept as current as possible.

Another concern for statistics collection is that of network shutdown. Whenever VTAM varies an NCP inactive, that NCP delivers statistics for all devices connected to it. If network shutdown is effected using the Z NET,QUICK command, VTAM varies all NCPs inactive, which then deliver their statistics. However, it is probable that Management Services is terminating because of the Z NET,QUICK command and therefore will not accumulate those statistics.

It is recommended that an orderly network shutdown be implemented whereby all NCPs are made inactive before VTAM is halted.

Customizing Device Configuration

You might consider customizing the configuration to include the following features:

- LPDA support
- RTM support
- FCS support

LPDA Support

If 386X type modems are used in the installation, then do this:

Step 1. Consider the inclusion of LPDA support.

Step 2. Set the LPDATS operand on the LINE macro to YES, or the solicitation of link status and DTE data from such devices will fail.

RTM Support

To be able to utilize the support NEWS provides for the 3x74 RTM feature, you must customize the controller for host support. To do this:

Step 1. During the customization process for the 3x74, select any one of the options available which provide host support. The specific option depends on your other requirements.

Step 2. Configure the default RTM definition and boundary values for all attached devices. These values can subsequently be changed by NEWS.

FCS Support

In order for NEWS to converse with a 3600/4700 controller, the controller must include the expanded System Monitor with Communications Network Management/Controller Support (CNM/CS). For further information on the generation statements required and CNM support, see the appropriate *3600/4700 Subsystem Instruction and Macros Reference*, *Programmer's Guide*, *System Programmer's Guide*, *Component Descriptions*, and *Principles of Operation* manuals.

8

Maintaining the NEWS Database

The NEWS Control Functions allow you to set NEWS parameters and tune the various features of NEWS after you have installed NEWS.

This chapter discusses the following topic:

- Maintaining the NEWS Database

Maintaining the NEWS Database

You can improve the capacity of the NEWS database by deleting database records or by reclaiming unused VSAM space.

The Database Maintenance panel allows you to delete specific records from the database, delete all records, or perform a manual reorganization of the database to reclaim VSAM space.

Enabling and Disabling CNMPROC Logging Options

The CNMLOGGING parameter group allows you to turn logging on and off for the NEWS database and to define what is to happen if the NEWS database is filled:

- Logging a reminder message after a specified number of lost records
- Whether or not to automatically reorganize the NEWS database

To implement the CNMPROC logging options:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the CNMLOGGING parameter group. The CNMLOGGING - NEWS Database Logging Options panel is displayed. This panel has two pages that you can scroll through to define CNMPROC logging options.

Step 3. On the first page, enter a value in the Logging Active? field if you want to suspend (NO) or resume (YES) logging.

Step 4. On the second page, enter a value in the Lost Record Reminder and Auto-reorg? fields, or leave the default values. These values specify what is to happen if the NEWS database is filled. For further information, press F1 (Help).

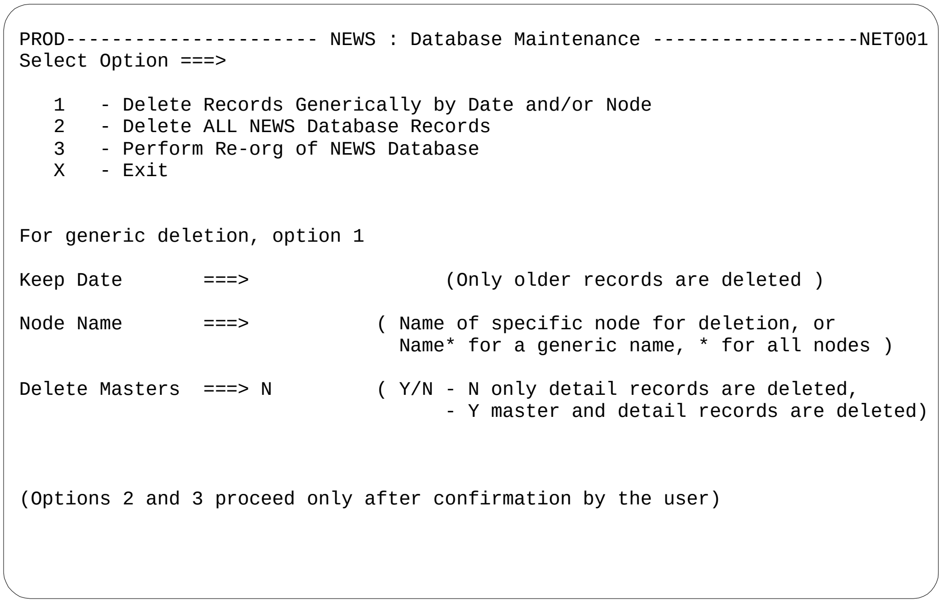
Step 5. Press F6 (Action) to action your entries.

Step 6. Press F3 (File) to save your settings.

Accessing Database Maintenance

To access Database Maintenance, enter **/SNADBA** at a ===> prompt. The NEWS : Database Maintenance menu is displayed.

Figure 8-1. NEWS : Database Maintenance Menu



```
PROD---------------------- NEWS : Database Maintenance ------------------NET001
Select Option ===>

   1   - Delete Records Generically by Date and/or Node
   2   - Delete ALL NEWS Database Records
   3   - Perform Re-org of NEWS Database
   X   - Exit


For generic deletion, option 1

Keep Date       ===>                 (Only older records are deleted )

Node Name       ===>           ( Name of specific node for deletion, or
                                 Name* for a generic name, * for all nodes )

Delete Masters  ===> N         ( Y/N - N only detail records are deleted,
                                     - Y master and detail records are deleted)



(Options 2 and 3 proceed only after confirmation by the user)
```

The panel allows you to do this:

- Choose an option from the following:
  - Delete specified records
  - Delete all records
  - Perform a reorganization of the NEWS database.
- If deleting specific records, specify the relevant parameters.

Deleting Records Generically by Date and/or Node

To delete specified records generically, from the NEWS : Database Maintenance menu, do this:

Step 1. Choose option **1 -** Delete Records Generically by Date and/or Node.

Step 2. Specify parameters for generic record deletion, by doing the following:

a. In the Keep Date field, specify a date to delete any records that arrived before the specified date. For details of date formats, press F1 (Help).

b. In the Node Name field, specify one of the following:

- A node name
- A generic node name, by typing a generic node name and the wild character *****
- All nodes, by typing the wild character *****

c. In the Delete Masters field, type **Y** to delete both master and detail records, or **N** to delete only detail records.

The Master record contains information, within a record category for a specific node, about when detailed records were collected, the record count, and the record collection period.

Detail records contain detailed information for a specific node.

A panel is displayed showing the number of detail records deleted, the number of master records updated, and the number of master records deleted.

> **Note**
>
> If you deleted large numbers of records, it is suggested that you perform a database reorganization to reclaim unused VSAM space. For information about how to do this, see the section, *Reorganizing the NEWS Database*, below.

Deleting All Records

To delete all records from the NEWS database, from the NEWS : Database Maintenance menu, choose option **2 -** Delete All NEWS Database Records. A confirmation message is displayed.

> **Note**
>
> After clearing the database, it is recommended that you perform a database reorganization to reclaim unused VSAM space. For information about how to do this, see the section, *Reorganizing the NEWS Database*, below.

Reorganizing the NEWS Database

Reorganizing the NEWS database allows you to reclaim any unused VSAM space.

Note

NEWS can perform automatic database reorganization if you enable the Auto Re-org facility by using the CNMLOGGING - NEWS Database Logging Options panel in ICS. For information about how to do this, see the section, *Enabling and Disabling CNMPROC Logging Options*, on page 8-2.

Before you reorganize the database, ensure you have appropriate VSAM definitions. To do this:

Step 1. Review the VSAM cluster definition of NEWSFILE on the NEWS - NEWS File Specifications panel in ICS.

Ensure that the NEWSFILE is defined with the REUSE option. For some levels of VSAM, this is only possible where the dataset has been sub-allocated.

Step 2. Review the backup file (NEWSBKP) definition on the NEWS - NEWS File Specifications panel in ICS.

Ensure that the backup dataset is large enough to contain all database records.

Step 3. Ensure that the NEWSFILE file ID has been freed by all procedures. To do this, go to OCS (=**O**) and enter SHOW UDBUSERS to check that the NEWSFILE is not being used.

For more information, see the section, *Releasing the NEWSFILE from CNMPROC*, below.

Releasing the NEWSFILE from CNMPROC

The NEWSFILE file ID would normally be in use by CNMPROC and any users currently using NEWS options involving access to the database. Certain options, such as the System Support Services menu, are entered without allocating the file until a specific requests required its use.

You can release the NEWSFILE from CNMPROC by suspending database logging, using the CNMLOGGING parameter group in ICS. For information about how to do this, see the section, *Enabling and Disabling CNMPROC Logging Options*, on page 8-2.

Manually Reorganizing the Database

To manually reorganize the NEWS database, from the NEWS : Database Maintenance menu, choose option **3 -** Perform Re-org of the NEWS Database.

Note

This action invokes NCL procedure $NWCNMRO which builds IDCAMS control statements and calls the utility program UTIL0007 to attach IDCAMS and perform the actual reorganization.

If the reorganization is successful, then a message is displayed to notify you of this. When this happens, restart CNMPROC or reactivate database logging by using the CNMLOGGING - NEWS Database Logging Options panel in ICS. For information about how to do this, see the section, *Enabling and Disabling CNMPROC Logging Options*, on page 8-2.

If the reorganization is not successful, then a message is displayed to notify you of this. When this happens, determine the reason for the failure by referring to the message issued to the activity log.

9

Tailoring NTS

Unless tailored, NTS collects data according to a set of default parameters. While these defaults are normally adequate for smaller networks, they might not suit your system. Failure to tailor these defaults, for larger systems in particular, is likely to result in NTS collecting far too much data, or the wrong mixture of data.

It is therefore highly recommended that you look at the particular needs of your installation and consider tailoring NTS to achieve the desired level and mix of data collection.

This chapter discusses the following topics:

- Defining NTS Classes
- Setting NTS System Parameters
- Establishing Inter-System Routing Links

Defining NTS Classes

Processing performed by NTS is determined by *class definitions*.

Within a given network, there are various different types of sessions and resources. You are likely to require specific types and amounts of data to be collected by NTS for each different session type, and to require different forms of processing for different session types. You are also likely to want to map session data to the underlying resource hierarchy.

You can achieve these objectives by defining four types of NTS classes to suit your installation needs. These classes are:

- Session classes
- Resource classes
- SAW classes
- RTM classes

By default, only SAW data is collected, and for *all* sessions, which might not suit your installation. No accounting, RTM, or resource statistics data can be collected until you have defined your classes.

For a discussion on NTS classes, see the section, *Using NTS Classes*, on page C-1.

Specifying the DEFCLASS Command

To define the attributes of the various categories or classes of session that control how NTS is to collect and process data, use the DEFCLASS command. (See the *Management Services Command Reference* manual, for a complete list of the attributes used to set up the class definitions.)

To define classes do this:

Step 1. Decide on what classes you need, and the attributes each class should have.

Step 2. Specify the DEFCLASS SESSION, RESOURCE, SAW, and RTM commands, and appropriate operand values in your NetMaster for SNA initialization procedure (normally $NSINIT).

Step 3. Periodically review the data collected by NTS and adjust any class definitions to suit new requirements.

To subsequently add class definitions, issue a DEFCLASS command from OCS. You must enter all operands, except those which have default values. If you do this while NetMaster for SNA is running, the new class definitions do not affect any existing sessions NTS is aware of but will be used by any new sessions.

Defining Session Classes

Session class definitions perform a dual function. They provide:

- The session selection criteria that determine to which session class each session belongs
- The names of SAW and RTM classes from which the member sessions derive their SAW and RTM class values

Each session class definition contains the following information:

- A unique session class identification name
- Parameters that a session must match, to be considered a member of this session class:
  - Full and partial names of primary and secondary resources
  - The subarea and APPN COS (Class-Of-Service) name for the session
  - An explicit route number and a virtual route number
  - A subarea and an APPN transmission priority
  - The session type and class
  - The source of the session
  - An SSCP name (identifying the domain of origin of the session)
- The names of the SAW and RTM class definitions that can be used by sessions in this class

Valid characters for operands within session classes include:

* Can be used in any position to represent a single wild character.

> Can be used as a suffix to indicate one or more trailing wild characters.

- Can be used as a suffix (for an LU only) to indicate one or more trailing wild characters for LU names that are *not* to be displayed.

Note

If any session class selection operands are omitted, any value of the omitted parameter is considered to be valid. For example, if no PRI operand is specified, any primary name is considered to be valid.

Table 9-1. Valid Values for the DEFCLASS SESSION Command Operands

Operand	Values	Associated Action
SAWCLASS	=sawclass	Sessions take their SAW class attributes from this class.
RTMCLASS	=rtmclass	Sessions take their RTM class attributes from this class.
PRI	=name	Names the primary resources considered to be in this session class.
SEC	=name	Names the secondary session partner for sessions considered as being in this session class.
COS	=cosname	Specifies the cosname of sessions considered as being in this session class.
APPNCOS	=cosname	Specifies the APPN cosname of sessions considered as being in this session class.
ER	=0-7	Provides the Explicit Route number (0 - 7) that sessions must have for them to be considered as being in this session class.
VR	=0-7	Provides the Virtual Route number (0 - 7) that sessions must have for them to be considered to be in this session class.
TP	=0-2	Provides the Transmission Priority number (0 - 2) that sessions must have for them to be considered to be in this session class.
APPNTP	=0-3	Provides the APPN Transmission Priority number (0 - 3) that sessions must have for them to be considered to be in this session class.
SCLASS	=SD =XD =XN	Provides the class of session as SD (same domain), XD (cross domain), or XN (cross network) that sessions must be for them to be considered as being in this session class.
STYPE	=LL =SL =SP =SS =MAI =CC	Provides the type of session as LL (LU-LU), SL (SSCP-LU), SP (SSCP-PU), SS (SSCP-SSCP), MAI (Multiple Application Interface), or CC (CP-CP) that sessions must be in for them to be considered as being in this session class.
SOURCE	=LOCAL =REMOTE =ALL	Provides the source of the session as LOCAL (sourced from VTAM on this system) or REMOTE (sourced from an ISR link with another NTS system), or ALL (sourced from either local or remote system).
SSCP	=sscpname	Valid only if SOURCE=REMOTE was specified. Provides the name of the SSCP at the system where a session was sourced.

Example

An example of a session class definition is shown and explained below.

```
DEFCLASS     SESSION=TSOB  PRI=TSO>  SEC=ASYD>
             SAWCLASS=NOLOG  RTMCLASS=TSO
```

In this example, the session class is called TSOA. For this class:

- Members are *primary* resources with a name that commences with the letters TSO, and *secondary* resources with a name that commences with the letters ASYD.
- Members use SAW class NOLOG, which specifies that session data be retained, but not logged.
- Members use RTM class TSO, which is defined on page 9-10.

Using Generic Names for Logging

All information logged to the NTS database is session-related and is stored under the session partner names. Together, the two network-qualified session partner names form a *session name pair.*

To limit the number of session name pairs stored in the NTS database, your session class definition parameters can specify generic session names (or part names), where possible.

For example, an application such as TSO has many ACB names that all begin with a common prefix, TSO>. This means that different sessions between a terminal and various TSO ACBs can all be logged under the same session pair name (that is, TSO>).

Defining Resource Classes

Resource class definitions determine the way NTS processes information for different network resources or groups of resources.

Resource class definitions contain the following information:

- A unique resource class identification name
- Parameters that resources *must* match (potentially: specific link, PU, or LU names) to be considered members of the resource class

> **Note**
>
> At least one of the following must be specified per resource definition: a LINK, PU, or LU name.

- Whether accounting statistics are to be collected

- A limit range (from 0 to 255) for the number of intervals that can occur before the statistics for the oldest interval are overwritten
- The names of the RTM class definitions that can be used by resources in this class

Valid characters for operands within resource classes include:

*	Can be used in any position to represent a single wild character.
>	Can be used as a suffix to indicate one or more trailing wild characters.
-	Can be used as a suffix (for an LU only) to indicate one or more trailing wild characters for LU names that are *not* to be displayed.

If you specify parameters other than the parameter that defines the level of the resource class, this has the effect of limiting the range of resources that match the resource class definition.

For example, if you specify both the PU and the LU parameters in the same resource class definition, the range of matching LUs is narrowed to those *owned* by the nominated PU(s). Because all resources should have unique names, this level of detail is only worthwhile if the value of the hierarchically lowest parameter in the class definition is generic, for example: LU=TSO>.

Table 9-2. Valid Values for the DEFCLASS RESOURCE Command Operands

Operand	**Values**	**Associated Action**
LINK	=*name*	Provides the full or partial link name that must be used by resources that are to be considered as being in this resource class.
PU	=*name*	Provides the full or partial PU name that must be used by resources that are to be considered as being in this resource class.
LU	=*name*	Provides the full or partial LU name of any LUs that are to be considered as being in this resource class.
STATS	=YES =NO	Provides the resource accounting statistics collection option for resources in this class.
LIMIT	=0-255	Valid only when STATS=YES is specified. Specifies in minutes (0 to 255) the interval to occur before the statistics for the oldest interval are overwritten.
RTMCLASS	=*rtmclass*	Specifies the RTM class name from which resources are to take their RTM class attributes if RTM summarization is required for this class.

Examples

Two examples of resource class definitions are shown and explained below.

```
DEFCLASS      RESOURCE=ALLINK  LINK=>  STATS=YES
              RTMCLASS=CICS

DEFCLASS      RESOURCE=TSO  LU=TSO>  STATS=YES
```

In the first example, the resource class is called ALLINK. For this class:

- All links are considered to belong to this class, as indicated by the specification LINK=>.
- Statistics are to be collected for members of this class.
- Members use RTM class CICS, which is defined on page 9-7. The format of RTM responses received by a resource are compared to the format defined in this RTM class definition, and statistics kept when a match is found.

In the second example, the resource class is called TSO. For this class:

- All LUs that have names starting with the letters TSO are considered to be members of this class (LU=TSO>).
- Statistics are to be collected for members of this class.
- Because no RTMCLASS parameter is specified, no RTM statistics will be collected; accounting statistics are, however, still collected.

Defining SAW Classes

Each SAW class that you define to NTS describes a set of processing options for all session awareness information, including whether such information is to be retained or discarded. SAW classes can therefore be used to ensure that no unwanted session data is collected, thereby saving both processing time and storage space. (See the examples on page 9-8.)

SAW class definitions contain the following information:

- A unique SAW class identification name
- Whether accounting statistics are to be collected
- Whether EDS system events are to be generated
- Whether session records are to be kept
- Whether NTS data is to be logged, and under what conditions
- The depth of the initial and final trace queues

Table 9-3 shows the available operands for the DEFCLASS SAW command, and the valid values for each operand. Default values are underscored.

Table 9-3. Valid Values for the DEFCLASS SAW Command Operands

Operand	Values	Associated Action
ACCT	=YES =NO	Accounting data is accumulated for this class. No accounting data is accumulated for this class.
EVENT	=YES =NO	Generates $$NTS.*xxx* events. Does not generate events.
KEEP	=YES =NO =LOCAL	Keeps data for this class. Does not keep data for this class. Sends data to a remote NTS.
LOG	=SUMMARY =DATA =ERROR =ALL =NO	Logs all data (except for trace data) at normal end of session; if session ends in error, all data, including trace data, is logged. Logs all data if any exists, otherwise logs no session data. Logs all data if session ends in error. Logs all data. Does not log any data.
TRACE	*=(n,n)*	Sets depth of the initial and final trace queue (default is 4,20).

Examples

Two examples of SAW class definitions are shown and explained below.

```
DEFCLASS     SAW=KEEP         ACCT=YES  KEEP=YES  LOG=ALL
                                         TRACE=(4,20)
DEFCLASS     SAW=NOKEEP       KEEP=NO
```

In the first example, the SAW class is, aptly, called KEEP. It specifies that:

- SAW data for sessions with which this class is associated is to be retained by NTS (KEEP=YES).
- Accounting data is to be accumulated for sessions with which this class is associated (ACCT=YES).
- All session and SAW data is to be unconditionally logged (LOG=ALL).
- The trace queue depth is to be restricted to 4,20 (that is, 4 PIUs in the initial queue and 20 in the final queue).

Only associate the type of sessions, for which you specifically wanted to retain *all* data, with this SAW class.

In the second example, the SAW class is called NOKEEP.

Because KEEP=NO is specified, NTS discards SAW data for any sessions with which this class is associated. This means that no information about these sessions is available, and no other NTS information can be collected for such sessions. Therefore, there is no point in specifying other operands for this class. This is a handy way of avoiding the collection of unwanted session data.

Defining RTM Classes

For NTS to be able to collect RTM information from network control units, you need to define one or more RTM classes. In addition, your control units (which might be 3274s, 3174s, or compatible devices) must have the required RTM hardware or microcode level support for the collection of RTM data, and have a host-modifiable RTM definition configured.

RTM class definitions contain the following information:

- A unique RTM class identification name
- Objective response times for this class
- Percentage of overall responses that must meet the objective response time for this class. Together, the values mean, for example, 90% of responses will be 1.5 seconds or less.
- Collection boundaries to be set in the control unit
- Definition criteria, to indicate what RTM data is to be kept

When NTS receives a session for which RTM data is to be collected, the boundary values for that class are set in the control unit, and retained for the duration of the session.

The objective response times and objective percentage for the class are used to monitor network response times, and can lead to the automatic generation of attention messages.

Table 9-4 shows the available operands for the DEFCLASS RTM command, and the valid values for each operand. Default values are underscored.

Table 9-4. Valid Values for the DEFCLASS RTM Command Operands

Operand	Values	Associated Action
OBJTIME	*=mm:ss.t*	Specifies the acceptable response time for the session. Range is from 0.1 seconds to 30 minutes. Can also be specified as *mm:ss, ss,* or *ss:t* (where *t* is a tenth of a second). This value must correspond to one of the boundary values.
OBJPC	*=1-100*	Specifies the objective percentage for this class.
BOUNDS	*=(value 1, value 2, ... value 4)*	Specifies up to four boundary values that are to be set in the control unit. One of the boundary values must be the same as the *objtime*.
RTMDEF	=FIRST =KEYBD =CDEB =LAST	Response time measured until the first character of the host data stream is received. Response time measured until the keyboard is unlocked. Response time measured until an SNA Change Direction or End Bracket order is received. Response time measured until the last character of the host data stream is received.

Examples

Two examples of RTM class definitions are shown and explained below.

```
DEFCLASS      RTM=CICS  OBJTIME=1.0  OBJPC=90
              BOUNDS=(0.5, 1.0, 2.0, 5.0)
DEFCLASS      RTM=TSO  OBJTIME=2.0  OBJPC=80
              BOUNDS=(1.0, 2.0, 5.0, 10.0)  RTMDEF=CDEB
```

In the first example, the RTM class is called CICS. For this class:

- The objective response time for the session is one second (OBJTIME=1.0).
- The objective percentage for this RTM class is 90 percent (OBJPC=90).
- Four boundary values are to be set in the control unit and used to accumulate RTM data for each session using this class {BOUNDS=(0.5,1.0,2.0,5.0)}.

Because the RTMDEF operand is not specified, response time is measured until the first character of the host data stream is received (this is the default).

In the second example, the RTM class is called TSO. For this class:

- The objective response time for the session is two seconds (OBJTIME=2.0).
- The objective percentage for this RTM class is 80 percent (OBJPC=80).

- Four boundary values are to be set in the control unit and used to accumulate RTM data for each session using this class (BOUNDS=(1.0,2.0,5.0,10.0)).
- Response time is to be measured until an SNA change direction or end bracket order is received (RTMDEF=CDEB).

Setting NTS System Parameters

NTS system parameters are used to:

- Define the NTS environment to VTAM.
- Specify global data collection options.
- Optimize NTS performance characteristics.
- Enable and disable data collection interfaces.
- Enable and disable NTS outputs.

Specifying the SYSPARMS Command

As well as enabling and disabling certain NTS functions by setting system parameters, you can set or modify certain system values by using the SYSPARMS command. This enables you to improve or modify NTS operations to suit your installation requirements. In most cases, the default values supplied by NTS should be adequate.

For further information about how NTS actually uses the values set by the system parameters, see Chapter 3, *How NTS Works*.

The NTS functions or parameter settings that you can tailor and the associated SYSPARMS operands are listed in the table below.

Table 9-5. SYSPARMS Operands for Tailoring NTS

Tailorable Function or Setting	Related Operand
NTS ACB name	NTSACBNM
Collection of accounting data	NTSACCT
Collection of resource statistics	NTSRSTAT
Logging of active sessions at shutdown	NTSCLOSE
Intensive message logging	NTSINTSV
Notification of MAI sessions	NTSMAISV
Generation of NTS events	NTSEVENT
Queuing of NTS CNM requests	NTSCNMQ
Consolidation of trace final queue buffers when first wrap occurs	NTSTRBFX
Presentation of MAI sessions to the NTS user exit	NTSMAIEX
Correlation of data	NTSCINTV
Trace activity	NTSMAXTR NTSMAXTP
Session keep counts	NTSSKEEP
VTAM session and trace data buffers	NTSSAWBF NTSTRCBF
Resource statistics collection intervals	NTSRSINT NTSRSLIM

A full description of the SYSPARMS command syntax, plus the valid operand values and their significance, can be found in the *SYSPARMS Operands* appendix of the *Management Services Administrator Guide*. For information about how to tailor NTS by using these operands to enable or disable functions, see the following sections in this chapter.

Defining the NTS ACB Name

You must have defined the NTS ACB name before you can use NTS. This task is normally performed as one of the NetMaster for SNA startup tasks, using the SAW parameter group in ICS. For details, see the *Unicenter NetMaster Network Management for SNA Implementation Guide*.

Collecting NTS Session Accounting Data

To enable the collection of NTS session accounting data, use the NTSACCT operand. Collection of this data can be either *selective* (the default) or *global.*

When session awareness processing is active, you can only specify NTSACCT=NO. To change to any other value, do this:

Step 1. Stop session awareness.

Step 2. Make the required modification.

Step 3. Restart session awareness.

Selective Accounting

NTSACCT=SELECTIVE is set by default. This means that accounting data for a session is only collected if the DEFCLASS ACCT operand is set to YES in the SAW class definition associated with the session.

If you require NTS accounting data to be collected for a particular session class, do this:

Step 1. Define an appropriate SAW class, with ACCT=YES specified.

Step 2. Associate this SAW class with the session class, by specifying the SAW class as the value for the DEFCLASS SAWCLASS operand in the session class definition.

Global Accounting

If you enable or disable the accounting function *globally,* SAW class definition accounting options are ignored.

To enable or disable the accounting function *globally*, specify NTSACCT=ALL or NTSACCT=NO.

Collecting NTS Resource Statistics

To enable the collection of NTS resource accounting and RTM statistics, use the NTSRSTAT operand. This operand globally enables or disables resource statistics collection when you specify a value of YES or NO. The default is NO.

Also consider these actions:

- Because NTS resource statistics are derived from session accounting data, ensure that NTSACCT=ALL is specified if you want resource accounting data to be collected.
- There is a hierarchy governing statistics collection for resources:
  - Collection must be enabled for the link used by a PU before statistics can be collected for the PU itself.
  - Collection must be enabled for the owning PU before statistics can be collected for an LU.
- If you specify NTSRSTAT=YES, but you do not want statistics collected for certain types of resources, specify STATS=NO in the resource class definitions for those types of resources.
- If the resource statistics function is globally disabled, the NTS resource class statistics collection option is ignored.
- When session awareness processing is active, you can only disable resource statistics collection.

To change from NTSRSTAT=NO to NTSRSTAT=YES, do this:

Step 1. Stop session awareness.

Step 2. Make the required change.

Step 3. Restart session awareness.

Caution

Carefully evaluate the requirements of your installation for resource statistics collection, because summarizing far too many resources may not give useful results and may adversely impact the performance of NTS.

For a description of how to set resource collection intervals, see the section, *Setting Resource Statistics Collection Intervals*, on page 9-19.

Rules Governing Statistics Collection

The rules governing statistics collection are:

- To collect statistics for an LU, statistics collection must be enabled for the PU that owns the LU.

- To collect statistics for a PU, statistics collection must be enabled for the link used by the PU.

- Accounting statistics for resources above the LU level are summarized from statistics collected for resources directly below them in the hierarchy. That is, PU accounting statistics are derived from statistics collected for LUs owned by the PU. Link accounting statistics are derived from statistics accumulated for the PUs that use the link.

Where a resource class specifies that statistics are to be collected, NTS accumulates resource statistics for resources that match the class, *at the level of the resource class*. For example:

- If the resource is an LU, and the selected resource class is at the LU level, statistics are collected for the specified LU *in isolation*.

- If the resource is an LU, and the selected resource class is at the PU level, statistics are collected for the specified LU, and are added to statistics collected from peer LUs owned by the same PU, to form the statistics for the owning PU. Statistics are not retained for the LU alone.

Monitoring NTS Resource Availability

If you have enabled the collection of statistics for a particular resource, NTS automatically uses SAW data to monitor the availability of that resource. A resource is considered to be *available* if it is participating in a session with the SSCP of the domain in which it is defined.

If NTS is monitoring resource availability, it automatically passes SMF records that indicate changes in the status of a resource to the NTS User Exit, if you have defined one.

Logging of Active Sessions at Shutdown

When Management Services is being shut down, all NTS activities must cease. It is highly likely that a number of sessions will still be active at this point, and that session data collected by NTS for such sessions will not have been logged.

To enable these residual sessions to be treated as ended for the purpose of logging, set the NTSCLOSE operand of the SYSPARMS command to YES. The sessions are queued for output processing, and the NTS class definitions checked to determine whether or not logging is actually required.

Because there is only a small delay (approximately 10 seconds) between the time that NTS is notified of the impending shutdown, and the actual termination of Management Services, this setting is useful only when the residual session count is small.

An alternative method of closing sessions is available through the SAWARE STOP CLOSE command, described in Chapter 10, *Collecting NTS Data*.

Note

To improve performance during NTS logging, operate the NTS database using VSAM Local Shared Resources (LSR) and deferred I/O capabilities. For information about space allocation for this VSAM dataset, see Appendix I, *NTS Storage Estimates*.

Enabling Intensive Message Recording

NTS receives large quantities of data from VTAM and may not process the data that it cannot understand. For example, NTS cannot collect trace data indefinitely for a session of which it has no knowledge. As a result, at some stage it purges such data. At other times, NTS might receive data that is not in the expected format and this data is discarded. During normal operation, these kinds of data are purged on a regular basis and might go unreported by NTS.

If you suspect that data is missing, to aid problem detection, you can enable intensive message recording to see if NTS is discarding any data.

To enable intensive message recording, set the NTSINTSV operand to YES. This causes log messages to be created whenever the conditions of data inconsistencies arise.

Enabling MAI Sessions

To enable NTS to be aware of MAI sessions, specify NTSMAISV=YES (the default is NO).

For trace and accounting data to be available for an MAI session, collection must be requested for the primary half-session component of an MAI virtual session. When trace data is received for the primary half session, and you have requested trace or accounting data collection for MAI sessions, NTS indicates that the MAI session has such data available. If you request the display of either trace or accounting data for an MAI session, primary or secondary, then the data collected for the primary half session is displayed.

Enabling NTS Session Event Generation

To enable or disable NTS event generation, do this:

Step 1. Use the NTSEVENT operand.

Step 2. Ensure that appropriate SAW classes (with EVENT=YES specified) are defined for and associated with sessions for which events are to be generated. (See *System Event Generation*, on page 3-8.)

Setting the Data Correlation Interval

One of the primary functions of NTS is to gather data from a number of sources and correlate it at session level.

The sequence in which data arrives, and the interval between such arrivals, is beyond the control of NTS. Under certain circumstances, such as a network failure, anticipated data might not arrive at all.

To protect NTS from waiting indefinitely for such session data, there is an interval defined that represents the time limit for data correlation. The default correlation interval is 30 seconds.

To change the default correlation interval, use the NTSCINTV operand.

It is recommended that the length of the correlation interval be kept constant throughout the network.

Setting Trace Limits

To impose limits on the number of specific trace requests that can be outstanding, use the NTSMAXTR operand.

To limit the number of PIUs that can be queued for any given session, use the NTSMAXTP operands. For information about the implications of restricting tracing, see Chapter 11, *Maintaining NTS*.

Setting Session Keep Counts

The session keep count refers to the number of session incidences that are stored concurrently in the NTS database for any session name pair. The default session keep count is 10.

To modify the default count, use the NTSSKEEP operand.

Note

The session keep count is only used the first time a session incidence for a new name pair is written to the database. The value is subsequently stored with the records in the database. To modify this value, use the NTSDBMOD command.

Setting VTAM Session and Trace Data Buffer Allocations

When NTS session awareness processing begins, requests are sent to VTAM specifying the number and size of the buffers to allocate for the collection of session awareness and session trace data.

To modify these values, use the NTSSAWBF and NTSTRCBF operands.

Default Allocations

By default, NTS allocates the following, which should be adequate for normal usage:

- Two buffers of 4K each to accommodate the flow of session awareness data from VTAM
- Four buffers of 4K each for the collection of session trace data

During times of exceptionally heavy trace activity, however, the allocation might be insufficient.

When These Allocations are Insufficient

If NTS cannot service the data buffers quickly enough, then VTAM overwrites the data in the oldest, unprocessed buffer, with the result that you lose data. NTS can detect this data loss and notify operators by issuing a monitor message.

In times of intense system activity, you might lose some trace data in this way, otherwise this kind of data loss is unlikely. (If it occurs, it needs investigation.)

Overcoming the Problem

If buffer overrun conditions occur, allocate a larger number of *smaller* buffers, rather than a smaller number of larger buffers. If more buffers are available to VTAM, they are likely to be available to NTS at any given time.

Other factors influence the delivery of data to NTS, especially the ability of the operating system to dispatch data. If its dispatching priority is too low, it might never be able to service large amounts of trace data in times of intensive activity. You need to check the dispatching priority, and ensure that it is set just below that of VTAM.

Setting Resource Statistics Collection Intervals

Resource statistics are collected and presented by NTS as counts of events that occurred within a specified time interval. Statistics gathered during different intervals can be compared for the purpose of network performance monitoring and analysis.

The valid range for the resource collection interval is 1 through 480 minutes (8 hours), and the default is 30 minutes.

To tailor the duration of the interval, use the NTSRSINT operand to set the value you require.

To set the value of the number of intervals that can occur before NTS overwrites the statistics collected for the oldest interval, use the DEFCLASS RESOURCE LIMIT operand, or set the global default by using the SYSPARM NTSRSLIM operand. The valid range of values for this operand 0 to 255; the default is 16.

Caution

High settings for this operand can consume large amounts of storage.

Establishing Inter-System Routing Links

A large installation can have NetMaster for SNA running on many systems active in many domains, and on a number of hosts.

To establish communication links between these systems, use the ISRIN and ISROUT parameter groups in ICS. For details, see *Enabling Multi-system Support*, on page 15-13.

Configuring Your Systems for NTS-SI

If you are planning to use NTS-SI for single network image presentation, a star network configuration of NTS systems is recommended. This enables you to either centralize or distribute the monitoring of network activity. How you configure ISR links also determines whether or not SAW and session data flows between systems, and in which directions. See Appendix C, *About the Session Awareness (SAW) Interface*, for a description of NTS-SI, and the SAW, and session data sharing rules.

Linking SNA Sessions

In an SNA environment, extra configuration data relating to cross-network sessions is made available to the host that controls the SNA gateway. This data includes network addresses and route information for sessions in the adjacent network.

To make the most effective use of NTS in an SNA environment, you must run NTS on the gateway host. You then link other host systems in the network to the Gateway host NTS system by establishing ISR links between these systems. This ensures that NTS has maximum accessibility to all session data.

Distributing MAI Data Across ISR

You can implement the NTS and MAI features in different domains on the same host. To route MAI data across ISR, ensure that the following conditions are met:

- NTS is licensed in both domains.
- SYSPARMS NTSMAISV=YES is specified in both domains.
- The NTS ISR link is configured for unsolicited message flow *from* the domain where MAI is resident, *to* the domain in which NTS is active.

10

Collecting NTS Data

Before you can productively use NTS, you need to activate session awareness processing. If session awareness processing has not been activated, no NTS data is collected. The only NTS function then available is the review of historical information in the NTS database.

This chapter discusses the following topics:

- Opening the VTAM CNM Interface
- Enabling NTS Session Awareness
- Connecting and Disconnecting the NTS Database

Opening the VTAM CNM Interface

NTS uses the CNM interface to transmit various commands to VTAM and to receive solicited and unsolicited data from VTAM. You must open the CNM ACB before you can start session awareness processing.

Note

The CNM interface is primarily used by NEWS.

To open the CNM ACB, use the CNM parameter group in ICS.

Enabling NTS Session Awareness

Before NTS can process session awareness data, it must establish a session with VTAM.

After a session is established between NTS and VTAM, VTAM sends session start notifications for all currently active sessions to NTS. This is the start of NTS session awareness, and is termed a *warm start.* From this point, VTAM sends session start and session end notifications to NTS as they occur for as long as session awareness remains active.

Starting and Stopping Session Awareness Processing

You use the SAW parameter group in ICS to activate session awareness processing in either of two ways:

- Automatically at initialization
- At any time after initialization

Connecting and Disconnecting the NTS Database

For historical recording purposes, NTS session awareness data can be logged to the NTS database.

The SAWLOG parameter group in ICS allows you to stop and start SAW logging on an ad-hoc basis, without needing to stop and start normal SAW processing.

Connecting the NTS Database

To ensure that NTS performs SAW logging:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the SAWLOG parameter group. The SAWLOG - Session Awareness (SAW) Logging panel is displayed.

Step 3. In the Logging Active? field, enter **Yes** to start logging SAW records at any time after initialization. For further information, press F1 (Help).

Step 4. Press F6 (Action) to action your entries.

Step 5. Press F3 (File) to save your settings.

Disconnecting the NTS Database

To stop logging SAW records:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the SAWLOG parameter group. The SAWLOG - Session Awareness (SAW) Logging panel is displayed.

Step 3. In the Logging Active? field, enter **No** to stop logging SAW records at any time after initialization. For further information, press F1 (Help).

Step 4. Press F6 (Action) to action your entries.

Step 5. Press F3 (File) to save your settings.

11

Maintaining NTS

Once installed and activated, NTS operates continuously without operator intervention. However, you might need to modify NTS processing from time to time in order to control its operation. For example, if a problem occurs, you might decide to record additional information, such as trace data.

This chapter discusses the following topics:

- Modifying NTS Processing
- Modifying NTS Class Definitions
- Limiting NTS Trace Activity
- Maintaining the NTS Database
- Writing NTS Records to SMF for Further Processing
- Controlling the MAI/NTS Interface

Modifying NTS Processing

After you have been using NTS for a time, you are likely to want to modify certain aspects of NTS processing. You would normally do this while session awareness is inactive, but you can also modify processing for active sessions, if necessary.

Modifying Processing While Session Awareness Is Inactive

You can modify NTS processing either by changing the **YES/NO** system parameter that determine which function NTS performs, or by specifying a parameter value.

To do this, use the SYSPARMS command.

Do this while session awareness is inactive. This ensures that the modifications apply to all session processing that occurs from the time you issue the SYSPARMS command with the appropriate operands and new values.

The NTS functions and settings that can be tailored using the SYSPARMS command and operands are listed in Table 9-5 on page 9-12.

Modifying Processing for Active Sessions

After NTS has built a session record for an active session, the future processing for that session is fixed by the various values extracted from the matching class or classes. However, you might need to modify such processing options under certain circumstances, especially since sessions can remain active for extended periods.

To modify the processing options for an active session, use the NTSMOD command.

Using the NTSMOD Command

The NTSMOD command enables you to do the following:

- Alter the trace queue depths; this can be useful, for example, if you want to collect more trace data for a session that is experiencing problems.
- Modify the NTS log options, to collect additional data for a session that is experiencing problems, by:
  - Logging all data when the session ends, regardless of the original SAW class log options
  - Forcing the current data to be logged in its present form, for future reference (a historical record)
  - Forcing the current session data to be presented to the NTS user exit instead of, or as well as, being logged to the NTS database

Example

This is an example of the use of the NTSMOD command.

```
NTSMOD NAME=CICS TRACE=(4,50) LOG=FORCE
```

In this case, the following occurs:

- The trace queue depth for sessions with the name CICS is modified.
- Sessions with the name CICS are flagged for force-logging and immediately placed on the output queue. The currently stored session data is logged, while normal NTS processing of the session continues.

When you review CICS session data at a later stage, the display of an **F** next to the end time for each session on the NTS Session List Panel indicates that these sessions were force-logged before they ended.

For more information about the NTSMOD command and its operands, see the *Management Services Command Reference* manual.

Hint

If you issue the NTSMOD command with neither the TRACE nor the LOG operand specified, then the sessions specified by the NAME operand are listed, so that you can determine the scope of the command prior to making any modifications.

Modifying NTS Class Definitions

After you have been using NTS for a time, you might want to modify one or more NTS class definitions. You would normally do this while session awareness is inactive.

Showing the Current Values of Parameters and Classes

To display current NTS class definitions, do this:

Step 1. Use the SHOW DEFCLASS command.

Step 2. Specify the type of class or classes you want to display—session, SAW, RTM, or resource—and, if you want to limit the display, the class name or partial name.

Example

This is an example of the use of the SHOW DEFCLASS command.

```
SHOW DEFCLASS RTM=CICS
```

In this case, one of the following occurs:

- If there is only one RTM class with a name starting with the letters CICS (or an entire name of CICS), then this is the only class listed.
- If there is more than one RTM class with a name starting with the letters CICS, then all these classes are listed.

Making the Required Changes

To replace or delete class definitions, use the REPCLASS and DELCLASS commands.

To change one or more attributes of a class, use the REPCLASS command to redefine the entire class. This command shares the same operands as the DEFCLASS command (see Chapter 9, *Tailoring NTS*).

Example

To delete the RTM class called CICS, issue the following command:

```
DELCLASS RTM=CICS
```

Limiting NTS Trace Activity

The STRACE command is used to start and stop global or specific NTS tracing. This command provides operands that enable you to select the precise session trace activity that you require. For a description of these operands, see the *Management Services Command Reference* manual.

Global tracing consumes large amounts of system resources. In order to avoid this, NTS provides parameters to limit the number of outstanding trace requests.

The PIU operand of the STRACE command allows you to do a PARTIAL or FULL tracing. In most instances, PARTIAL tracing provides sufficient data for problem determination.

Caution

> Before you issue a request to trace a complete RU from VTAM, take note that an RU can be very large.

Limiting the Number of Concurrent Traces

To set the maximum number of specific trace requests that can be outstanding at any time, specify the limit in the value of the SYSPARMS NTSMAXTR operand.

This value includes the following requests:

- Specific trace start requests (even if these are pending)
- When global tracing is active, specific trace stop requests
- Any specific trace requests started by the NTS selective accounting function, which operate automatically if you specify ACCT=YES in a SAW class definition.

NTS rejects any attempt to issue a specific trace request that would result in the value set for NTSMAXTR being exceeded.

Limiting Trace PIU Collection

To set the trace queue depths for the initial and final trace queues for a session, set the values in your SAW class definitions. These values determine the maximum number of PIUs that can be stored for a session at any given time.

To override the values in the SAW definition, issue the SYSPARMS NTSMAXTP command.

Note

> When you have defined values for NTSMAXTP, you cannot define a new SAW class with trace queue depths exceeding these values, nor change the trace queue depths of an existing SAW class to be greater than the values set for this system parameter.

Maintaining the NTS Database

To maintain the NTS database, use the NTSDBMOD command. This command enables you to:

- Delete session records.
- Alter session keep counts for sessions stored in the NTS database.
- Cancel the execution of a previously issued NTSDBMOD command.

Example

This is an example of the use of the NTSDBMOD command:

```
NTSDBMOD PRINAME=CICS KEEPDATE=2001/03/31
```

In this case, all stored records for sessions with the primary name of CICS that predate April 1, 2001, are deleted.

For more information about the NTSDBMOD command and its operands, see the *Management Services Command Reference* manual.

Modifying the Database Session Keep Counts

After the first session incidence has been recorded for a session name pair, the master record contains the session keep count for that name pair.

To display this value, use the SHOW SKEEP command.

To modify this value, use the NTSDBMOD command.

To delete all records for the session name pair, including master and cross-reference records, set a new session keep count of zero in the NTSDBMOD command.

The NTSDBMOD command allows a generic name specification to permit mass update and deletion with a single command.

Writing NTS Records to SMF for Further Processing

You can convert NTS session records to SMF type 37 format to use the output in report-generating applications.

To do this, you must specify the name of an NTS user exit in the SAW parameter group in ICS.

NTS automatically passes SMF-formatted records to this exit. The supplied exit can be customized to perform further processing of the data before the data is passed to SMF and finally to a report-generating application, such as SAS, to produce statistical reports based on the raw NTS data.

For a description of the formats used by NTS, see Appendix G, *NTS SMF Record Formats*.

Controlling the MAI/NTS Interface

To ensure that MAI notifies NTS of all currently existing MAI sessions, specify SYSPARMS NTSMAISV=YES when NTS and MAI are already active.

If you specify NTSMAISV=NO when the interface is already active, NTS retains knowledge of existing MAI sessions, but is not notified of any new MAI sessions.

12

Tailoring NCS

This chapter discusses the following topic:

- Limiting the Size of Display Lists and Individual Node Displays

Limiting the Size of Display Lists and Individual Node Displays

When NetMaster for SNA is running under an XA or ESA operating system, all of the storage used by NCS is above the 16 Mb line and therefore has no effect on the private region size required to run Management Services.

When NetMaster for SNA is running under a non-XA operating system or when paging rates are of significant concern, you may want to limit the amount of storage used by NCS when processing summary display lists.

Limiting the Number of Lines in NCS Display Lists

The maximum size of any summary display list within NCS is 9999 entries. A list of this size requires a considerable amount of storage in the NetMaster for SNA private region.

Limiting the Number of Subnodes Displayed For a Node

In a Fujitsu VTAM-G environment, VTAM display commands support an optional NUMBER= operand which can be used to limit the number of subnodes for a particular node.

Implementing NCS Display Limits

To set a maximum number of lines in NCS display lists and a maximum number of subnodes displayed for a node, use the SNA Node Display Limits parameter group in ICS:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the NCS parameter group. The NCS - SNA Node Display Limits panel is displayed.

Step 3. Enter the Maximum Number of Display Lines. This value controls the number of resources displayed in NCS.

Step 4. If you are using Fujitsu VTAM-G, enter the Maximum Number of Sub-nodes. This value controls VTAM display command results.

Step 5. Press F6 (Action) to action your entries.

Step 6. Press F3 (File) to save your settings.

13

Tailoring and Controlling NCPView

This chapter discusses the following topics:

- About NCP Monitoring
- Defining a System Image
- Defining NCP Resources
- Tailoring the NCPView NCL Exit

About NCP Monitoring

NCP monitoring enables you to:

- View performance information about the NCPs in your network
- Perform diagnostics on selected NCPs

Before you can use the NCP monitor, you must have a system image that defines the resources you want to monitor. You can specify whether or not performance monitoring is done for each resource defined in the system image.

Performance monitoring uses data sampled at regular intervals. The information retrieved by data sampling is used to:

- Trigger alerts if the monitored performance is outside defined boundaries
- Generate online reports that can be viewed from the NCP monitor

System Images

The system image represents the set of resources you can monitor and control. Each system image has a name and a version number. You can define multiple system images, but only one system image can be active in a region at a time. The system image becomes active when it is loaded. A system image is loaded in either of the following ways:

- At region startup
- By issuing the LOAD command

Your product region uses a default system image if no system image is successfully loaded during startup.

During system image load:

- Performance monitoring is started for the NCPs defined in the system image.
- The NCPs are defined to the NCP monitor.

NCP Definitions

NCP definitions are qualified by:

- The system image name and version
- The NCP name

Working with NCPs

The NCP administration facilities allow you to:

- Update, copy, and delete the NCPs defined when you implement your product region
- Add new resources
- Define which NCPs are monitored
- Set the attributes to be monitored for each NCP
- Define the conditions that cause alerts to be raised and actions taken

Monitoring Resources in a Multisystem Environment

In a multisystem environment, you can view and perform diagnostics from a single monitor on the resources from the connected systems.

In a multisystem environment, each region must load a different system image. Each NCP's system image name is visible on the NCP monitor. For subordinate regions, the system image name must match the name supplied during the multisystem linking process.

Defining a System Image

If you do not want to use the default system image, you can define a system image to your product region. To do this:

Step 1. Enter **/RADMIN.I** at the ===> prompt. The System Image List panel is displayed. This panel lists the system images defined to your system. You can:

- Add new system image definitions
- Browse, change, copy, and delete existing system images

Step 2. To add a new image, press F4 (Add). The System Image Definition panel is displayed.

Step 3. Specify the name of the system image, its version, and a short description of the system image in this panel.

One system image is required for each region. If you are defining a system image for a subordinate, use the name assigned during the multisystem linking process.

Step 4. Press F3 (File). You are returned to the System Image List panel, and a message is displayed indicating that the system image has been successfully added to the knowledge base.

Defining NCP Resources

Your existing NCP resources are automatically defined when you implement your NetMaster for SNA region. After implementation you can use the resource definition facility to:

- Update, copy, or delete existing definitions
- Add new resource definitions

Note

For any NCPs that you wish to monitor, ensure that the value specified for the OPTIONS keyword in the NCP SYSCNTRL definition statement is STORDSP. This value allows storage information to be displayed by NCPView. See the IBM *NCP, SSP, and EP Definition Reference* manual.

To define a new NCP resource to be monitored by the NCP monitor:

Step 1. Enter **/RADMIN.R** at the ===> prompt. The ResourceView : Resource Definition panel is displayed. This panel displays the system image name and lists the resource classes that you can maintain.

Step 2. Enter **S** in front of the NCPMON (NCP Monitor) class. The ResourceView : NCP Monitor List panel is displayed. The NCP resources already defined to the system image are listed on this panel.

You can use this panel to:

- Add new NCP resource definitions
- Browse, update, copy, and delete existing NCP resource definitions

Note

Alternatively, you can perform these functions from the NCP Monitor panel:

- Press F4 (Add) to add new NCP resource definitions.
- Enter DB beside a resource to browse or update its definition.

Step 3. Press F4 (Add). The NCP Monitor General Description panel is displayed.

Step 4. Enter the NCP Monitor Name. This defines the NCP resource to your system.

Step 5. Set Monitoring to Active and provide a description of the NCP resource.

Step 6. If you want to use one of the predefined templates for the monitoring definition, enter **L** in the Template Name selection field.

Step 7. Press F8 (Forward). The NCPMON Monitoring Definition panel is displayed.

Step 8. Set the frequency for monitor samples to be taken in the Monitor Interval field—this can be from 5 to 60 minutes.

Step 9. Press F10 (Attributes) to edit the attributes to be monitored.

Step 10. Press F8 (Forward). The NCPMON Automation Log Details panel is displayed. This panel defines the resource transient log.

It is recommended that you accept the default settings for this feature. For more information, press F1 (Help).

Step 11. Press F8 (Forward). The Owner Details panel is displayed. The fields on this panel are for documentation purposes only.

Step 12. If required, complete the fields on the panel and press F3 (Save). The NCP Monitor List panel is displayed with the new definition added.

Tailoring the NCPView NCL Exit

Distributed with NCPView is an NCL exit, called ZNCUX000, that you use to tailor NCPView functions. This exit is called at the end of NCPView initialization to allow you to include code specific to your installation. You can, for example, include code that:

- Filters out NCPs that you do not want monitored
- Allocates NCP unformatted dump files

Whenever NCPView finds an NCP, ZNCUX000 is called to determine whether the NCP should be included or excluded from NCPView's monitoring scope. This can happen during NCPView initialization, or whenever NCPView detects a new NCP being activated.

To tailor the ZNCUX000 NCL exit, follow these steps:

Step 1. Take a copy of ZNCUX000.

This is distributed in the *?dsnq*.SN400.SNTEXEC dataset.

Step 2. Place the copy in the TESTEXEC library.

Step 3. Change any of the following functions as required:

- Exclude one or more NCPs.
- Allocate NCP dumps.

Detailed instructions on how to change these functions are included as comments within the ZNCUX000 procedure.

Warning

Never delete the distributed ZNCUX000 exit.

14

Tailoring NetView Operator Command Emulation

The NetView operator command emulation facility assists former NetView users with the commands used in NetMaster for SNA.

This allows users to operate NetMaster for SNA by using the same commands and procedures they are accustomed to using with NetView.

This chapter discusses the following topic:

- Modifying Table Entries

Modifying Table Entries

The command emulation tables are contained in a CAS table called EQUATES. Each table entry represents a NetView operator command and can be defined as ACTIVE or INACTIVE.

You can modify the EQUATES table while NetMaster for SNA is running by using CAS Table Services:

Step 1. Enter **/CASTABE** at a ===> prompt. The CAS : Table Entry Definition Menu is displayed.

Step 2. Enter **L** at the ===> prompt, **$VW** in the Appl ID field, and **EQUATES** in the Field Name field. The CAS : Table Entry Definition List is displayed.

Step 3. Enter **U** next to the command that you want to change. The CAS : Table Entry Definition panel is displayed.

Step 4. Change only the Active? (Yes/No) field. See the following section, *Considerations When Modifying Table Entries*, for further information.

Step 5. Press F3 (File) to return to the CAS : Table Entry Definition Menu.

Step 6. Use option **R** to reload the table.

Step 7. Action the SNAINIT parameter group in ICS for the changes to take effect. See the section, *Identifying Your Initialization Procedure (SNAINIT),* on page 6-5.

Considerations When Modifying Table Entries

Any changes you make are limited by the following rules:

- Changing an entry from INACTIVE to ACTIVE will not set a global equate. To do this, you must restart NetMaster for SNA.
- Changing an entry from ACTIVE to INACTIVE will take affect as soon as the table is reloaded. When an entry is set to INACTIVE, the NetView operator command is no longer operational.

Note

When you inactivate NetView operator commands that have the same name as MS commands, that is ACT and INACT, the inactivation does not become effective until NetMaster for SNA is next restarted.

- If an entry is INACTIVE when the operator tries to use it, the system attempts to execute it as if it were an NCL procedure name, and an error message results—for example:

  ```
  START commandname
  N04005 NCL PROCEDURE commandname DOES NOT EXIST IN LIBRARY
  COMMANDS.
  ```

The NetView operator commands initialization process, described in the *Unicenter NetMaster Network Management for SNA Implementation Guide*, can be performed locally, that is, on entry to OCS. However, this means that:

- The equates only stay in effect while the operator remains in OCS.
- You cannot set an equate for a command that has the same name as a Management Services (MS) command, that is, the ACT and INACT commands.

15

Advanced Configuration Tasks

This chapter describes the tasks to configure the advanced features of NetMaster for SNA.

This chapter discusses the following topics:

- Implementing a Trouble Ticket Interface
- Implementing the Alert History Function
- Applying Alert Monitor Filtering
- Forwarding Alerts
- Enabling Multi-system Support
- Managing Focal Points
- Managing Entry Points
- Maintaining Control File Records
- Maintaining Resource Alias Names
- Allocating NCP Unformatted Dumps
- Integrating NCS with a Configuration Management Database
- Creating User Alerts

Implementing a Trouble Ticket Interface

The alert monitor provides a number of actions you can set up to run automatically when alerts arrive:

- Notify
- Execute a command
- Execute NCL
- Generate a trouble ticket

To use the trouble ticket action, you must implement a trouble ticket interface. The alert monitor supports two interfaces to trouble ticket systems.

- Electronic mail, where an e-mail describing the problem can be sent to a trouble ticket application or to a particular person. This method can be used to send problems to many types of trouble ticket applications.
- Custom, where you can write your own NCL code to deliver the trouble ticket to an application by whatever means you choose.

To implement a trouble ticket interface, you need to define the trouble ticket interface between your product region and the alert monitor (see the section, Defining a Trouble Ticket Interface).

If you want the operator to supply information when requesting creation of a trouble ticket, you also need to set up the trouble ticket data entry definition (see the section, Setting Up the Trouble Ticket Data Definition).

Defining a Trouble Ticket Interface

To define a trouble ticket interface between your product region and the Alert Monitor, do this:

Step 1. Enter **/ALADMIN** at the ==> prompt. The Alert Monitor : Administration Menu is displayed.

Step 2. Select option I – Define Trouble Ticket Interface. The Alert Monitor : Interface Definition panel is displayed.

Step 3. In the Interface Type field, specify the type of interface you want to define.

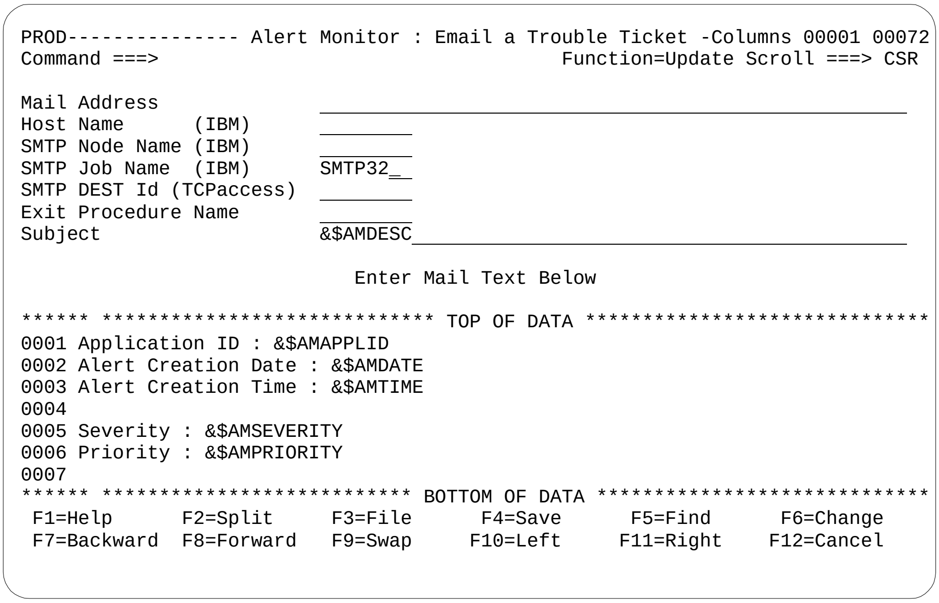
Enter a question mark (?) in this field to obtain a selection list of valid values.

Step 4. Press F6 (Action). A panel is displayed where you can define your trouble ticket interface. The type of panel displayed varies, depending on the interface type that you specified. See the sections, Defining an E-mail Trouble Ticket Interface, and Defining a Custom Trouble Ticket Interface, for further details.

Defining an E-mail Trouble Ticket Interface

If you specified EMAIL as the interface type on the Alert Monitor : Interface Definition panel, press F6 (Action) to display the Alert Monitor : Email a Trouble Ticket panel.

Figure 15-1. Alert Monitor : Email a Trouble Ticket Panel



```
PROD--------------- Alert Monitor : Email a Trouble Ticket -Columns 00001 00072
Command ===>                                   Function=Update Scroll ===> CSR

Mail Address              ___________________________________________________
Host Name      (IBM)      ________
SMTP Node Name (IBM)      ________
SMTP Job Name  (IBM)      SMTP32__
SMTP DEST Id (TCPaccess)  ________
Exit Procedure Name       ________
Subject                   &$AMDESC___________________________________________

                             Enter Mail Text Below

****** ***************************** TOP OF DATA ******************************
0001 Application ID : &$AMAPPLID
0002 Alert Creation Date : &$AMDATE
0003 Alert Creation Time : &$AMTIME
0004
0005 Severity : &$AMSEVERITY
0006 Priority : &$AMPRIORITY
0007
****** *************************** BOTTOM OF DATA *****************************
 F1=Help      F2=Split     F3=File      F4=Save      F5=Find      F6=Change
 F7=Backward  F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

To define your e-mail trouble ticket interface, do this:

Step 1. Enter values in the input fields in the top section of the panel.

Use F1 (Help) to obtain information about completing these fields.

Step 2. Complete the Enter Mail Text Below section of the panel, which is free format.

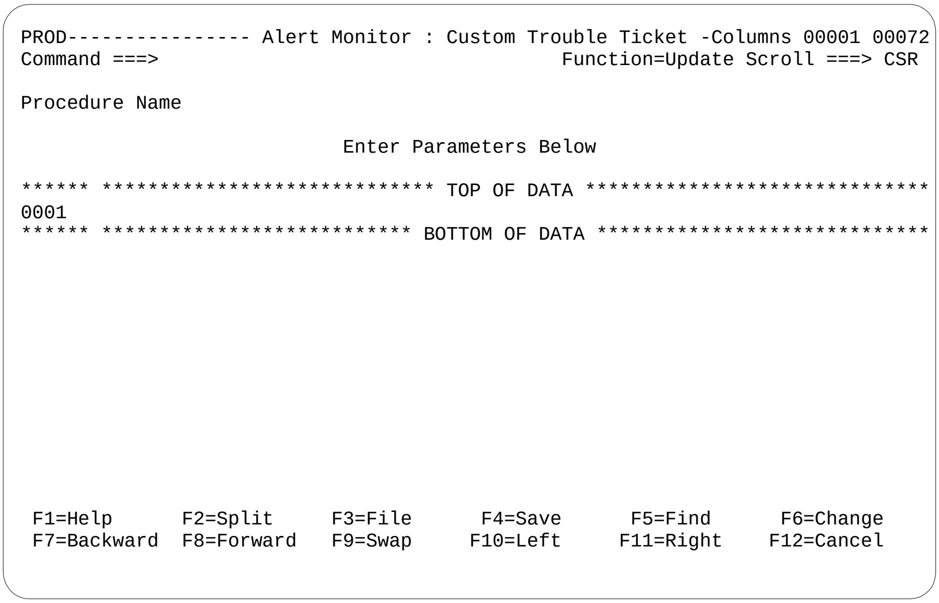
Use F1 (Help) to obtain information about completing this section.

Step 3. Press F3 (File). You are returned to the Alert Monitor Administration Menu and your interface definition is saved.

Defining a Custom Trouble Ticket Interface

If you specified CUSTOM as the interface type on the Alert Monitor : Interface Definition panel, press F6 (Action) to display the Alert Monitor : Custom Trouble Ticket panel.

Figure 15-2. Alert Monitor : Custom Trouble Ticket Panel



```
PROD---------------- Alert Monitor : Custom Trouble Ticket -Columns 00001 00072
Command ===>                                   Function=Update Scroll ===> CSR

Procedure Name

                            Enter Parameters Below

****** ***************************** TOP OF DATA ******************************
0001
****** *************************** BOTTOM OF DATA *****************************









 F1=Help      F2=Split     F3=File      F4=Save      F5=Find      F6=Change
 F7=Backward  F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

To define your custom trouble ticket interface, do this:

Step 1. In the Procedure Name input field, enter the name of your NCL procedure for delivering trouble tickets.

Step 2. In the Enter Parameters Below section of the panel, specify any parameters that you want the NCL procedure to receive. This section is free format.

Press F1 (Help) to obtain information about completing this section.

Step 3. Press F3 (File). You are returned to the Alert Monitor Administration Menu and your interface definition is saved.

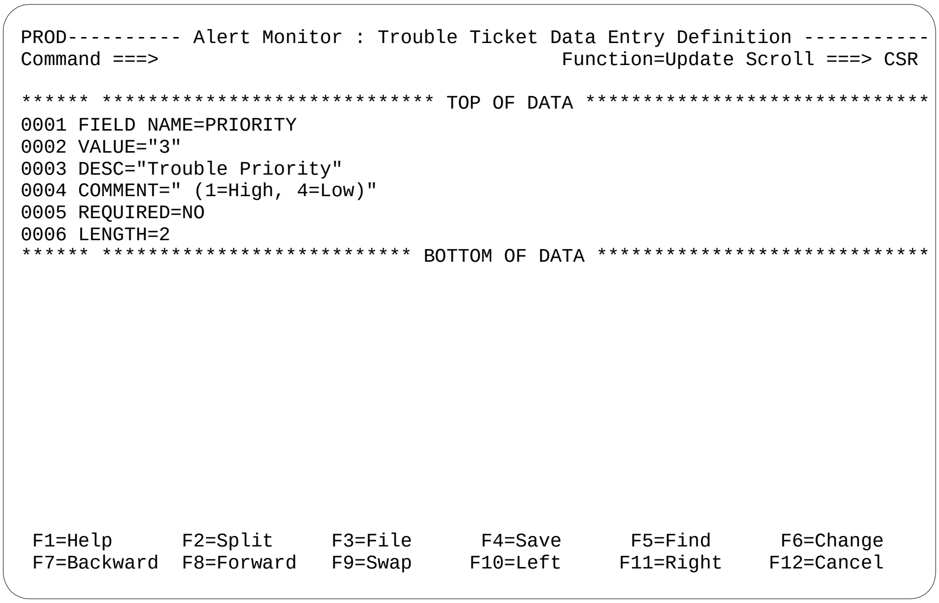
Setting Up the Trouble Ticket Data Definition

If you want the operator to supply information when requesting a trouble ticket, you need to set up the trouble ticket data entry definition.

To define the information you want the operator to supply, do this:

Step 1. On the Alert Monitor Administration Menu, select option **D -** Trouble Ticket Data Definition. The Alert Monitor : Trouble Ticket Data Entry Definition panel is displayed.

Figure 15-3. Alert Monitor : Trouble Ticket Data Entry Definition Panel



```
PROD---------- Alert Monitor : Trouble Ticket Data Entry Definition -----------
Command ===>                                   Function=Update Scroll ===> CSR

****** **************************** TOP OF DATA *****************************
0001 FIELD NAME=PRIORITY
0002 VALUE="3"
0003 DESC="Trouble Priority"
0004 COMMENT=" (1=High, 4=Low)"
0005 REQUIRED=NO
0006 LENGTH=2
****** ************************** BOTTOM OF DATA ****************************










 F1=Help      F2=Split     F3=File      F4=Save      F5=Find      F6=Change
 F7=Backward  F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Step 2. In the free-format data entry section of the panel, enter the data entry definition for the panel that the operator will use when creating a trouble ticket.

Press F1 (Help) to obtain information about completing this section.

Step 3. Press F3 (File). You are returned to the Alert Monitor Administration Menu and your trouble ticket data entry definition is saved.

Implementing the Alert History Function

The alert monitor retains data in an alert history file. To specify how long alerts are to be retained in this file, do this:

Step 1. Enter **/ICS** at the ===> prompt. The ICS : Customization Panel appears.

Step 2. Enter **U** in front of the $NM ALERTHIST parameter group. The Alert History File Specification details appear.

Step 3. In the Days to Retain Alerts in History File field, specify the number of days that you want alerts to be retained in the history file.

Step 4. In the Time of Day for Alert Purge field, specify the time of day (in the format *hh.mm*) at which the Alert Monitor will delete alerts that have been in the history file longer than the retain setting.

Step 5. Press F3 (File) to save your settings.

To reorganize the Alert History database to reclaim dead space, do this:

Step 1. Copy (REPRO) the History to a backup file.

Step 2. Delete and redefine the original file.

Step 3. Copy the data back in to the redefined file.

You should also monitor the amount of disk space used by the dataset, to estimate the optimal file size and optimal frequency of reorganization.

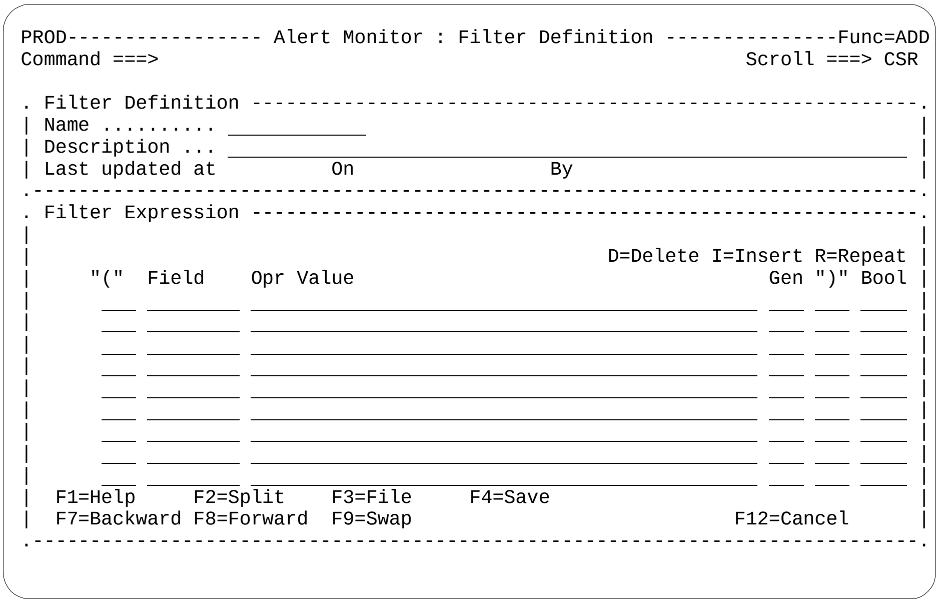
Applying Alert Monitor Filtering

You can filter the alerts raised by the alert monitor by applying a set of criteria to each of the fields within the alert. The filters that you create can be named and stored for later use. To apply filtering to the alerts that are raised, do this:

Step 1. Enter **/ALADMIN.F** at the ===> prompt. The Alert Monitor : Filter Definition List panel is displayed.

Step 2. To create a new filter press F4 (Add). The Alert Monitor Filter panel is displayed.

Figure 15-4. Alert Monitor : Filter Definition Panel



```
PROD----------------- Alert Monitor : Filter Definition ---------------Func=ADD
Command ===>                                                   Scroll ===> CSR

. Filter Definition ---------------------------------------------------------.
| Name .......... ____________                                              |
| Description ... ___________________________________________________________ |
| Last updated at          On                 By                            |
.----------------------------------------------------------------------------.
. Filter Expression ---------------------------------------------------------.
|                                                                           |
|                                                 D=Delete I=Insert R=Repeat |
|     "("  Field    Opr Value                                  Gen ")" Bool |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|      ___ ________ ____________________________________________ ___ ___ ____ |
|  F1=Help     F2=Split    F3=File     F4=Save                              |
|  F7=Backward F8=Forward  F9=Swap                          F12=Cancel      |
.----------------------------------------------------------------------------.
```

Step 3. Enter the name and description of the filter.

Step 4. Enter the values that you require into the Field, Opr, Value, Gen, and Bool fields. For a full description of these fields press F1(Help).

Step 5. Press F3 (File) to save the changes.

Forwarding Alerts

Alerts are normally displayed on the Alert Monitor display. However, you can also forward them to other platforms:

- UNIX platforms as SNMP traps
- NetMaster for SNA (NEWS) or NetView (TME10) systems, as generic alert NMVTs
- Unicenter Event Management

Note

To forward alerts to UNIX platforms as SNMP traps, or to Unicenter Event Management, you must implement the TCP/IP sockets interface, using the SOCKETS parameter group in ICS. See the chapter titled *Starting NetMaster for SNA for the First Time* in the *Unicenter NetMaster Network Management for SNA Implementation Guide*.

You can apply filter criteria to forward different types of alerts to different platforms.

Implementation

To enable alert forwarding, execute the $AMEVFWD command. This command has two groups of parameters:

- Those which describe the destination platform
- Those which supply the filtering criteria

To implement Alert Forwarding, you need to execute an $AMEVFWD command for each combination of platform and filtering parameters. It is recommended that you include these commands in your READY procedure so that they are enabled at initialization and remain in effect while the system is running.

See *$AMEVFWD,* on page 15-8, for the syntax of the $AMEVFWD command.

The SNMP Trap Definition

The MIB definition for alerts forwarded as SNMP traps is provided in member $AMTRAP, supplied in the dsnpref.MS500.INSTAL dataset. You can download this member to your UNIX system and compile it.

> **Note**
>
> When copying this member to your UNIX system, you can rename it to avoid problems on some UNIX systems where the $ sign has special meaning.

The supplied MIB defines two traps with the following object identifiers:

- $AMTRAP = 1.3.6.1.4.1.1126.1.2.1.2 (for an alert)
- $AMTRAPC = 1.3.6.1.4.1.1126.1.2.1.3 (when an alert is cleared)

$AMEVFWD

Function Implements alert forwarding.

Operands The destination platform is specified by the DESTTYPE, DESTADDR, DESTPORT, COMMNAME, and NETVALRT parameters. The filtering criteria are specified by the FILTER parameter. Only alerts that meet these criteria are forwarded.

Destination Details

Destinations can be TNG traps, SNMP traps, or (Generic Alert) NMVTs.

```
$AMEVFWD      DESTTYPE=TNGTRAP
               DESTADDR=ipaddr
             [ DESTPORT=ipport ]
             [ COMMNAME=community ]
             [ FILTER=filtername ]
             [ CLEAR={ YES | NO } ]
               ALERTS= { NEW | ALL }
             [ IPINACT={ WAIT | IGNORE } ]
```

DESTTYPE=TNGTRAP

Specifies that the alerts are forwarded as SNMP traps optimized for the Unicenter Event Console (TNGTRAP). It differs from SNMPTRAP as follows:

- Providing four fields only: system, product, severity, and text
- Putting the text field last
- Passing new alerts only

See the *dsnpref*.MS500.INSTAL($AMTRAP) MIB definition for a list of the fields (variables) that are present in the standard SNMP traps.

DESTADDR=*ipaddr*

Specifies the IP address to which alerts are to be forwarded. The address can be in dotted notation (for example, 123.45.6.78) or a host name address (for example, network.operations.com).

DESTPORT=*ipport*

Specifies a port number at the destination. The default is 162.

COMMNAME=*community*

Specifies the community name (at the destination address). Defaults to public (in lower case). Community names are case-sensitive. When a value is specified, ensure that the value is in the correct case.

If the procedure is started from another procedure, for example within the READY procedure, ensure that the statement remains case-sensitive at execution time. By default, NCL procedures such as the READY procedure convert data to upper case during assignment. Disable upper case translation over the $AMEVFWD procedure invocation with the &CONTROL UCASE|NOUCASE statement. For example:

```
&CONTROL NOUCASE
START $AMEVFWD DESTTYPE=SNMPTRAP DESTADDR=xxx +
DESTPORT=162 COMMNAME=public
&CONTROL UCASE
```

IPINACT=WAIT|IGNORE

Specifies what to do when SNMP traps cannot be forwarded because the TCP/IP interface is inactive:

- WAIT—suspends alert forwarding until the interface becomes active. This value is the default.
- IGNORE—discards the alerts when the interface is inactive.

Sending Alerts as SNMP Traps

```
$AMEVFWD      DESTTYPE=SNMPTRAP
               DESTADDR=ipaddr
              [ DESTPORT=ipport ]
              [ COMMNAME=community ]
              [ FILTER=filtername ]
              [ CLEAR={ YES | NO } ]
               ALERTS= { NEW | ALL }
              [ IPINACT={ WAIT | IGNORE } ]
```

DESTTYPE=SNMPTRAP

For SNMP traps, specific destination details are required, as follows:

DESTADDR=*ipaddr*

This is mandatory. It specifies the destination (IP) address as either a valid IP address in dotted notation (for example, 123.45.6.78) or a host name address (for example, network.operations.com).

DESTPORT=*ipport*

This is optional. It specifies the port number (at the destination address). If specified, it must be a number in the range 1 to 65535. If omitted, it defaults to 162.

COMMNAME=*community*

Specifies the community name (at the destination address). Defaults to public (in lower case). Community names are case-sensitive. When a value is specified, ensure that the value is in the correct case.

If the procedure is started from another procedure, for example within the READY procedure, ensure that the statement remains case-sensitive at execution time. By default, NCL procedures such as the READY procedure convert data to upper case during assignment. Disable upper case translation over the $AMEVFWD procedure invocation with the &CONTROL UCASE|NOUCASE statement. For example:

```
&CONTROL NOUCASE
START $AMEVFWD DESTTYPE=SNMPTRAP DESTADDR=xxx +
DESTPORT=162 COMMNAME=public
&CONTROL UCASE
```

IPINACT=WAIT|IGNORE

Specifies what to do when SNMP traps cannot be forwarded because the TCP/IP interface is inactive:

- WAIT—suspends alert forwarding until the interface becomes active. This value is the default.
- IGNORE—discards the alerts when the interface is inactive.

Sending Alerts as NMVTs

```
$AMEVFWD       DESTTYPE=NMVT
              [ NETVALRT=ppiname ]
              [ FILTER=filtername ]
              [ CLEAR={ YES | NO } ]
                ALERTS= { NEW | ALL }
```

DESTTYPE=NMVT

NMVTs are queued to the NETVALRT PPI receiver queue and are then processed by either NetMaster for SNA or NetView.

NETVALRT=*ppiname*

The default name for the NETVALRT PPI receiver is NETVALRT. If you have used an alternate name, use this operand to specify the alternate name.

Note

If you have also set up the PPINETVALRT parameter group in ICS to receive PPI events, you should not forward alerts to the NETVALRT PPI receiver.

Forwarding to NetMaster for SNA

To receive alerts, you must have the PPI receiver active. This is started on the destination NetMaster for SNA system.

The forwarded alerts are recorded as Operator Notification EVENTS for the specific resource, which is the primary system name (PRI=*initparm*) of the system that forwarded the alert.

Forwarding to NetView

To receive alerts in a NetView region you must have the CNMCALRT task defined and active. To do this:

Step 1. Check the DSIDMN (or DSIDMNB) member in the DSIPARM PDS.

Step 2. Ensure that the CNMCALRT task is included and is initialized (INIT=Y). For example:

```
TASK MOD=CNMCALRT,TSKID=CNMCALRT,PRI=6,INIT=Y
```

Note

This statement is necessary for the OS/390 software alert forwarding function.

The alerts are formatted as Operator Notification generic alerts.

Filtering Details

There are three options for filtering. You can:

- Apply a named filter using FILTER=filtername.
- Apply filtering criteria with the CLEAR and ALERTS operands—these can be applied, regardless of the destination details. Both are optional.
- Not apply a filter.

Applying a Named Filter

FILTER=*filtername*

This specifies the name of the filter to apply. If specified, the named filter must exist. For more information see Applying Alert Monitor Filtering.

Predefined Filter Criteria

CLEAR=YES|NO

This parameter relates to alerts that are cleared. If set to YES, alerts for cleared alert events are forwarded. This parameter is optional.

ALERTS=NEW|ALL

This parameter controls how alerts generated before alert forwarding is enabled are handled.

If set to NEW—only new alerts are forwarded. This option would normally be used if alert monitoring is restarted.

If set to ALL—current alerts, and new alerts as they arrive, are forwarded. Use this option to make the alert forwarding function behave the same as the Alert Monitor option at a terminal. This option would normally be used when the alert monitor is started during initialization of the system region.

Examples

To send all NCPVIEW alerts from the local system as SNMP traps, enter:

```
START $AMEVFWD DESTTYPE=SNMPTRAP +
DESTADDR=network.operations.com +
DESTPORT=4567 FILTER=NCPVIEW
```

To send all alerts from SOLVE5 as NMVTs, enter:

```
START $AMEVFWD DESTTYPE=NMVT SYSTEMID=SOLVE5
```

Enabling Multi-system Support

If you have product regions on different OS/390 images, you can link them together, using INMC links, to form a multi-system configuration.

A multi-system configuration enables you to log onto your local product region and view and control the resources of linked product regions. You can do things such as:

- Display a VTAM node in a remote region
- Display the alerts raised from all the linked regions
- Monitor NCP utilization in all the linked regions

Multi-systems are set up and administered from the Automation Services : Multi-System Support Menu. To access this menu, enter **A.M** at the ===> prompt of the Primary Menu.

For more information press F1(Help).

Caution

See the chapter, *Administering a Multi-system Environment*, in the *Automation Services Administrator Guide* before setting up a multi-system environment.

Defining Communications for NEWS and for NTS

The INMC facility allows both local and remote systems to be connected as network peers. When the INMC links are defined, you can define feature-specific communications.

Using INMC, the ISR (Inter System Routing) facility provides communication services between NEWS features and between NTS features in multiple systems.

To enable these services, define ISR network traffic between the peers by defining the ICS parameters ISRIN and ISROUT. You must define the ICS parameters only on the appropriate system, as shown in the following example:

Example

To send ISR traffic from system A to system B:

- Define ISROUT on system A, specifying system B’s link name
- Define ISRIN on system B, specifying system A’s link name

Defining ISR Inbound Parameters (ISRIN)

To define ISR inbound parameters, do this:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the ISRIN parameter group. The ISRIN - ISR (Inbound) panel is displayed.

Step 3. Specify link names for PPO messages, CNM data, and SAW data.

Step 4. Specify whether these links are to NetView (PPO messages and CNM data only).

Step 5. Press F6 (Action).

Step 6. Press F3 (File).

Defining ISR Outbound Parameters (ISROUT)

To define ISR outbound parameters, do this:

Step 1. Enter **/ICS** at a ===> prompt. The ICS : Customization Parameters panel is displayed.

Step 2. Enter **U** beside the ISROUT parameter group. The ISROUT - ISR (Outbound) panel is displayed.

Step 3. Specify link names for PPO messages, CNM data, and SAW data.

Step 4. Press F6 (Action).

Step 5. Press F3 (File).

ISR and NTS

In NTS, control is implied through the system and class parameter settings that request the type of data NTS is to collect.

For a description of the interaction between NTS and ISR, see the section, *Establishing Inter-System Routing Links*, on page 9-20.

ISR Environments

For a general description of ISR environments, see the *Management Services Administrator Guide*.

Managing Focal Points

The Focal Point Management menu options apply to APPN network nodes and allow authorized users to manage and maintain the definitions of the backup and nesting focal points for the Problem Management category of SNA Management Services (SNAMS). Managing these definitions ensures that Problem Management information from the APPN network flows to a centralized management focal point.

For details about SNAMS and focal point management, see the following IBM manuals:

- *SNA Management Services Reference*
- *SNA Transaction Programmer's Reference Manual for LU Type 6.2*

About Focal Points and Entry Points

In Advanced Peer-to-Peer Networking (APPN), roles are established through the interchange of SNAMS capabilities between two nodes. One of these nodes assumes the role of a *focal point*, the other becomes the *entry point*. When this exchange has been established, the entry point is said to come under the sphere of control of the focal point.

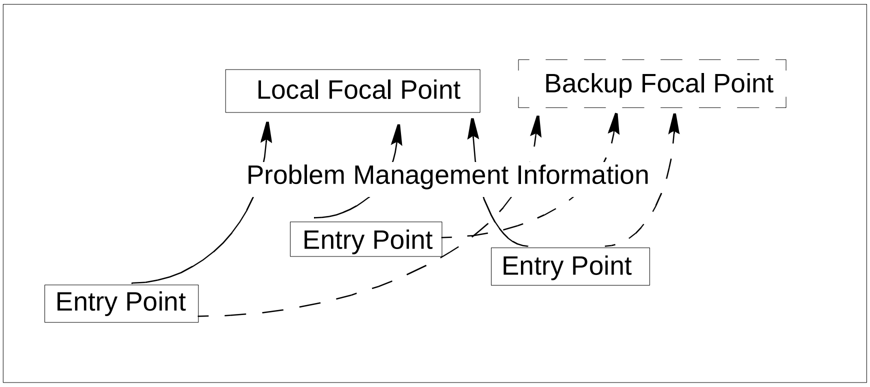
A focal point provides centralized management for one or more entry points under its sphere of control. Each entry point can only have one focal point, but the same focal point can provide services for multiple SNAMS categories.

Both focal points and entry points are dynamic. This means that if a primary focal point becomes unavailable, a *backup focal point* can be requested. This also means that higher ranked focal points can replace existing focal points.

About Local and Backup Focal Points

A *local focal point* is a focal point with entry points locally registered to it. A local focal point might become inactive after acquiring entry points. In this event, any Problem Management information from the entry points that the local focal point has acquired is sent to a *backup focal point*, shown in Figure 15-5.

Figure 15-5. Backup Focal Point

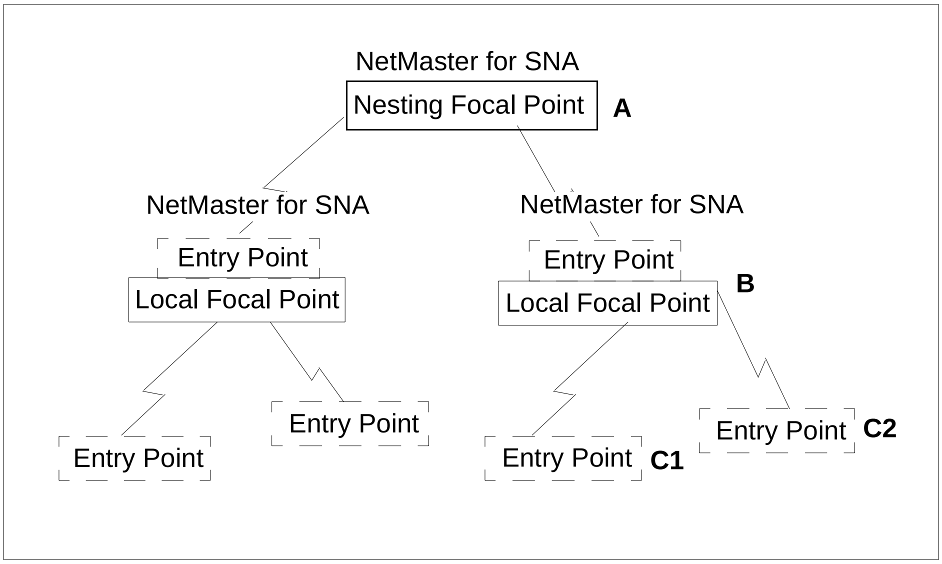


About Nesting Focal Points

One focal point can come under the control of another focal point. This is called *nesting*. Nesting is typically used where each focal point is managing a different level of SNAMS.

In an SNA environment, a nesting focal point, shown in Figure 15-6, is a focal point that has registered a local focal point as an entry point.

Figure 15-6. Nesting Focal Point

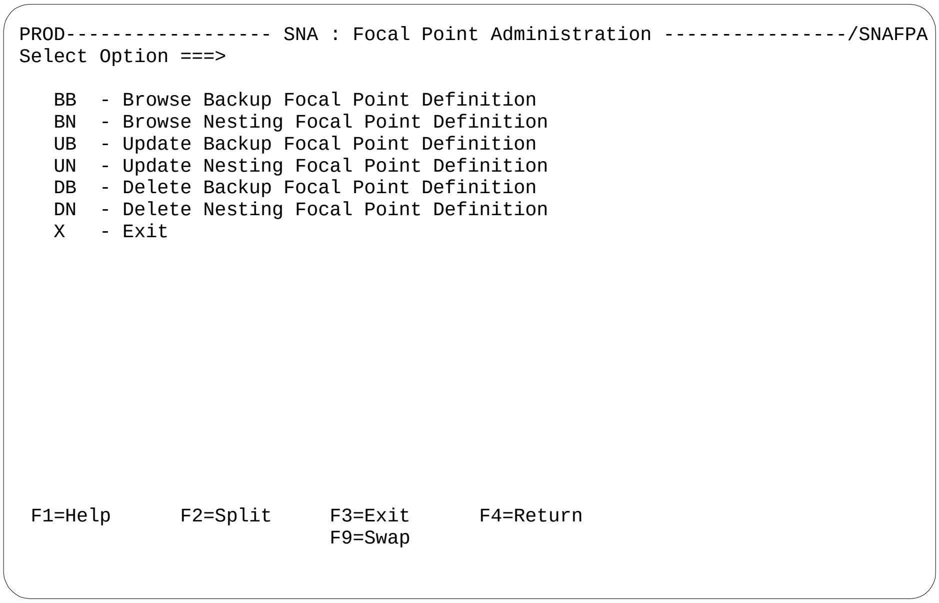


For example, in Figure 15-6, the local focal point B sends any Problem Management information, passed to B from entry points C1 and C2, on to the nesting focal point A.

Browsing, Updating, or Deleting Focal Points

To access the SNA : Focal Point Administration menu, enter **/SNAFPA** at a ===> prompt.

Figure 15-7. SNA : Focal Point Administration Menu



```
PROD------------------ SNA : Focal Point Administration ----------------/SNAFPA
Select Option ===>

   BB  - Browse Backup Focal Point Definition
   BN  - Browse Nesting Focal Point Definition
   UB  - Update Backup Focal Point Definition
   UN  - Update Nesting Focal Point Definition
   DB  - Delete Backup Focal Point Definition
   DN  - Delete Nesting Focal Point Definition
   X   - Exit












 F1=Help      F2=Split     F3=Exit      F4=Return
                           F9=Swap
```

To browse, update, or delete a backup focal point or a nesting focal point, type the relevant letters for the option you want at the ===> prompt.

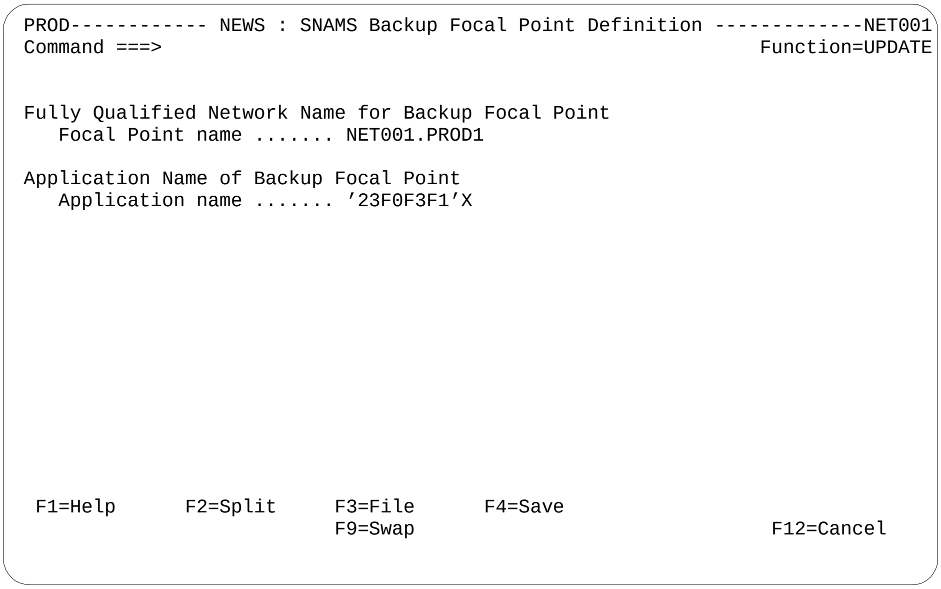
For details of these options, press F1 (Help).

If you choose a browse, update, or delete option for a backup focal point, the NEWS : SNAMS Backup Focal Point Definition panel, shown in Figure 15-8, is displayed.

If you choose a browse, update, or delete option for a nesting focal point, the NEWS : SNAMS Nesting Focal Point Definition panel, shown in Figure 15-9, is displayed.

Defining Backup Focal Points

Figure 15-8. NEWS : SNAMS Backup Focal Point Definition Panel



```
PROD------------ NEWS : SNAMS Backup Focal Point Definition -------------NET001
Command ===>                                                    Function=UPDATE


Fully Qualified Network Name for Backup Focal Point
   Focal Point name ....... NET001.PROD1

Application Name of Backup Focal Point
   Application name ....... ’23F0F3F1’X














 F1=Help      F2=Split     F3=File      F4=Save
                           F9=Swap                              F12=Cancel
```

The SNAMS Backup Focal Point Definition panel displays definitions for the SNAMS backup focal point.   For further details, press F1 (Help).

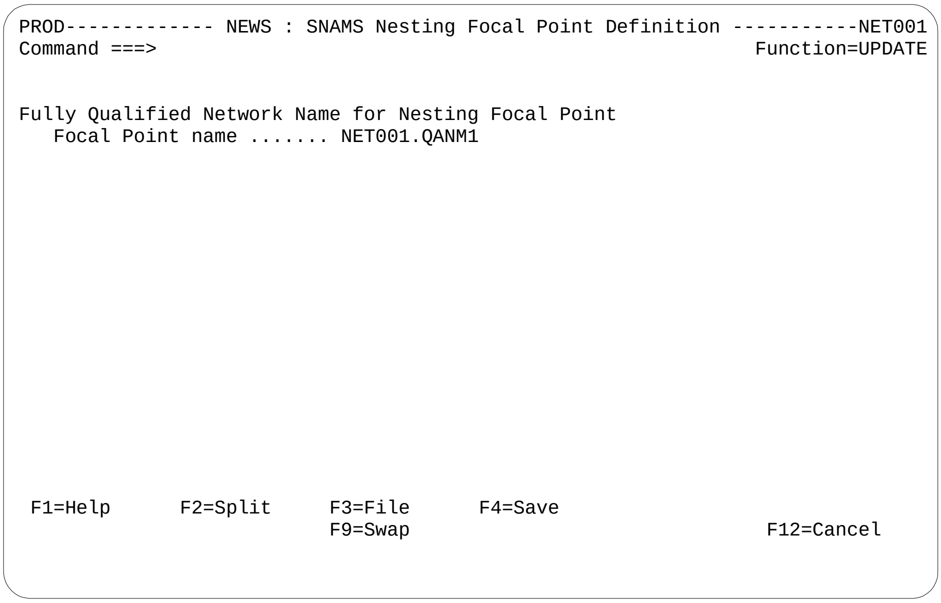
To define a backup focal point, do this:

Step 1. In the Focal Point Name field, specify a name in the form *Network Identifier* and *Network Addressable Unit* (NAU) separated by a period (for example, NTWKNAME.NAUNAME).

Step 2. In the Application Name field, type a four-byte hexadecimal quoted string (in the format ’*aabbccdd*’X”) if the string contains non-display characters.

Step 3. Press F3 (File).

Figure 15-9. NEWS : SNAMS Nesting Focal Point Definition Panel



```
PROD------------- NEWS : SNAMS Nesting Focal Point Definition -----------NET001
Command ===>                                                    Function=UPDATE


Fully Qualified Network Name for Nesting Focal Point
   Focal Point name ....... NET001.QANM1










 F1=Help      F2=Split     F3=File      F4=Save
                           F9=Swap                                F12=Cancel
```

The SNAMS Nesting Focal Point Definition panel displays the definition for the SNAMS nesting focal point. For further details, press F1 (Help).

To define a nesting focal point, do this:

Step 1. Specify a focal point name in the form *Network Identifier* and *Network Addressable Unit* (NAU) separated by a period (for example, NTWKNAME.NAUNAME).

Step 2. Press F3 (File).

Managing Entry Points

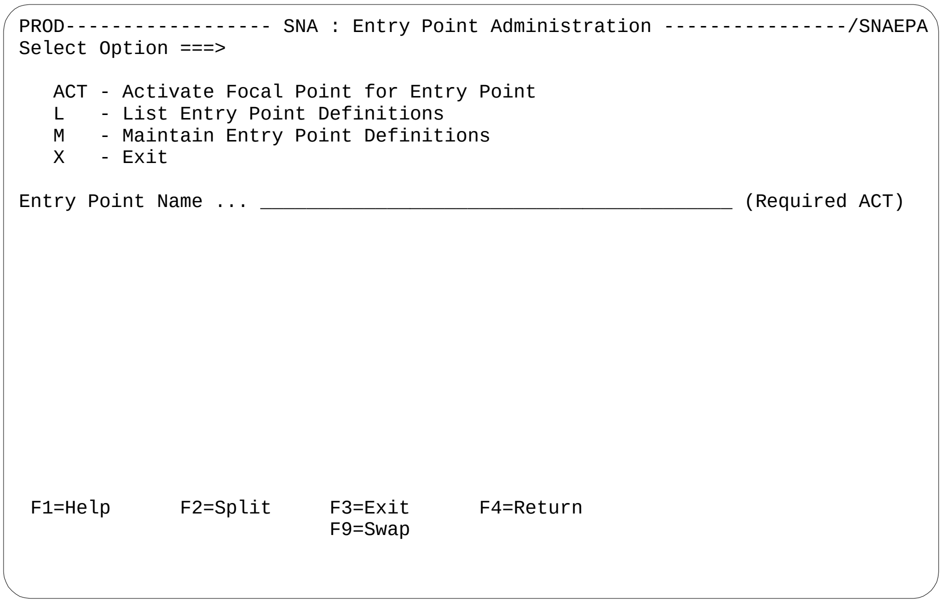
The SNA : Entry Point Administration menu allows authorized users to manage and maintain the definitions of the Entry Points for the Problem Management category of SNA Management Services (SNAMS). It also allows you to acquire entry points for the NetMaster for SNA focal point so that the focal point can receive Problem Management information from them.

For information about focal points and entry points, see the section, *About Focal Points and Entry Points*, on page 15-15.

From the SNA : Entry Point Administration menu, you can access the SNAMS EP Definitions panel to maintain entry point definitions for NetMaster for SNA, or get a full or partial list of currently defined entry points.

To access the SNA : Entry Point Administration menu, enter **/SNAEPA** at a ===> prompt. For information about the options on this menu, press F1 (Help).

Figure 15-10.SNA : Entry Point Administration Menu



```
PROD------------------ SNA : Entry Point Administration ----------------/SNAEPA
Select Option ===>

   ACT - Activate Focal Point for Entry Point
   L   - List Entry Point Definitions
   M   - Maintain Entry Point Definitions
   X   - Exit

Entry Point Name ... _________________________________________ (Required ACT)









 F1=Help      F2=Split     F3=Exit      F4=Return
                           F9=Swap
```

Activating a Focal Point

You can acquire an entry point for the NetMaster for SNA focal point by defining the entry point you want to acquire for the focal point. To do this:

Step 1. Type **ACT** at the ===> prompt on the NEWS : Entry Point Administration menu.

Step 2. In the Entry Point Name field, type a fully-qualified node name in the form *Network Identifier* and *Network Addressable Unit* (NAU) separated by a period (for example, NTWKNAME.NAUNAME).

If you enter only the NAUNAME portion of the name, NEWS prefixes it with the name of the network in which this system is active.

Step 3. Press ENTER. The SNAMS EP Definitions panel (see Figure 15-12) is displayed.

Maintaining Entry Point Definitions

The Entry Point Definitions panel displays a list of entry point definitions that can be registered to the Problem Management focal point of NetMaster for SNA. From the list, you can select an entry point definition to browse, update, or delete.

To list entry point definitions:

Step 1. On the Entry Point Administration menu, enter **M** at the ===>prompt.

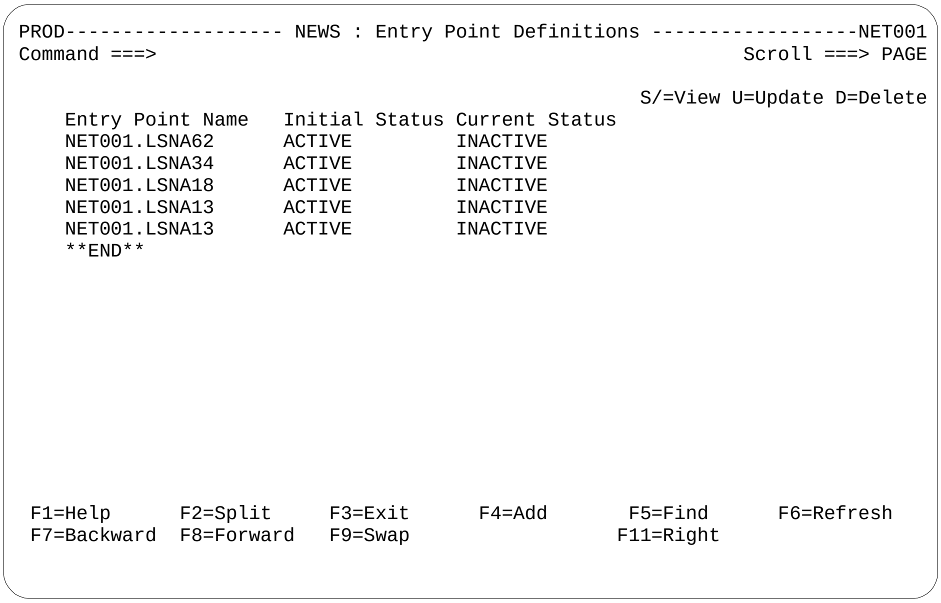
Step 2. (Optional) To narrow the range of entry points listed, specify a prefix in the Entry Point Name field. Only entry point names beginning with the entered prefix are listed.

Step 3. Press ENTER. The Entry Point Definitions panel is displayed, in update mode.

Note

If you just want to browse entry points, enter **L** (instead of **M**) on the Entry Point Administration menu. This displays the Entry Point Definitions panel in Browse mode.

Figure 15-11. NEWS : Entry Point Definitions Panel—Update Mode



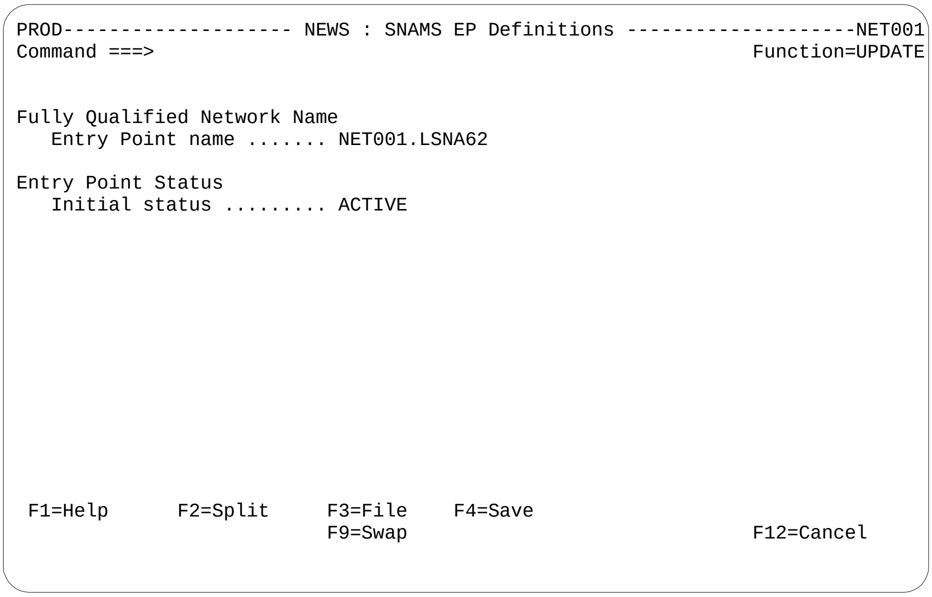
```
PROD------------------- NEWS : Entry Point Definitions ------------------NET001
Command ===>                                                   Scroll ===> PAGE

                                                        S/=View U=Update D=Delete
    Entry Point Name   Initial Status Current Status
    NET001.LSNA62      ACTIVE         INACTIVE
    NET001.LSNA34      ACTIVE         INACTIVE
    NET001.LSNA18      ACTIVE         INACTIVE
    NET001.LSNA13      ACTIVE         INACTIVE
    NET001.LSNA13      ACTIVE         INACTIVE
    **END**










 F1=Help      F2=Split     F3=Exit      F4=Add       F5=Find      F6=Refresh
 F7=Backward  F8=Forward   F9=Swap                  F11=Right
```

Updating an Entry Point

To update an entry point from the Entry Point Definitions panel:

Step 1. Enter **U** beside an entry point name on the list. The SNAMS EP Definitions panel is displayed for the selected entry point.

Figure 15-12.NEWS : SNAMS EP Definitions Panel—Update Mode



```
PROD-------------------- NEWS : SNAMS EP Definitions --------------------NET001
Command ===>                                                     Function=UPDATE


Fully Qualified Network Name
   Entry Point name ....... NET001.LSNA62

Entry Point Status
   Initial status ......... ACTIVE












 F1=Help      F2=Split     F3=File    F4=Save
                           F9=Swap                            F12=Cancel
```

Step 2. Change the details as required.

Step 3. To file the changes, press F3. To save the changes, press F4.

Defining an Entry Point

To define an entry point from the Entry Point Definitions panel:

Step 1. Press F4 (Add). The SNAMS EP Definitions panel is displayed.

Step 2. In the Entry Point Name field, type a name in the form *Network Identifier* and *Network Addressable Unit* (NAU) separated by a period (for example, NTWKNAME.NAUNAME).

Step 3. In the Initial Status field, type **ACTIVE** or **INACTIVE**.

Step 4. To file the changes, press F3. To save the changes, press F4.

Deleting an Entry Point

To delete an entry point from the Entry Point Definitions panel, enter **D** beside an entry point name on the list.

Maintaining Control File Records

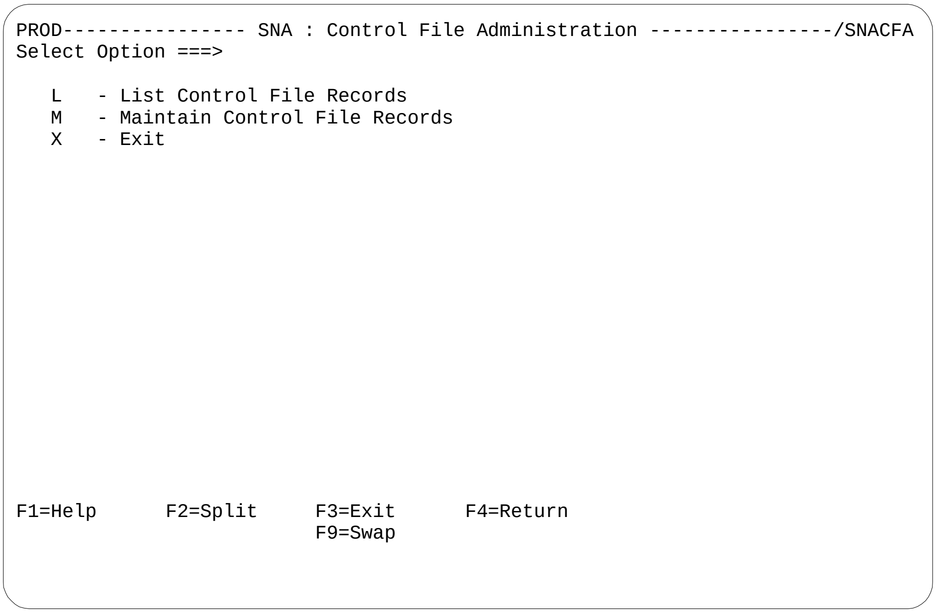
You can tailor the processing of NEWS facilities by maintaining the Network Services Control File (also called the NSCNTL database). The Control File contains records that control NEWS processing and provide a database of messages for the display of solicited and unsolicited records about network events. NEWS uses the control file to determine the CNM processing path for solicited and unsolicited records.

The NEWS : Control File Category Maintenance panel allows you to browse, modify, or add existing support for Control File records of a specified category.

Accessing the Control File Category Maintenance Panel

To maintain the Control File records, enter **/SNACFA** at a ===> prompt. The SNA : Control File Administration menu is displayed.

Figure 15-13.SNA : Control File Administration Menu



```
PROD---------------- SNA : Control File Administration ----------------/SNACFA
Select Option ===>

   L   - List Control File Records
   M   - Maintain Control File Records
   X   - Exit
















F1=Help      F2=Split     F3=Exit      F4=Return
                          F9=Swap
```

This panel allows you to list and maintain control records. For further information, press F1 (Help).

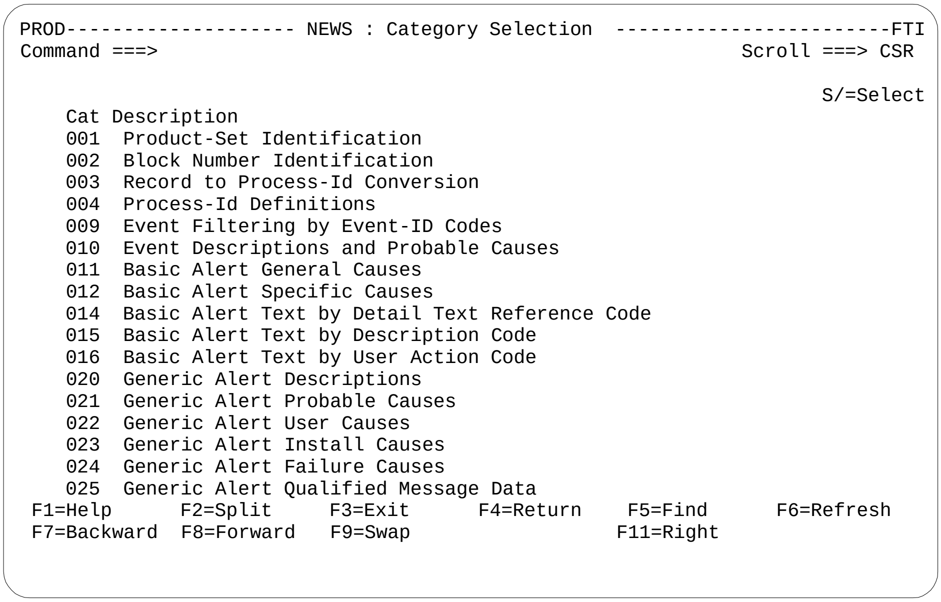
Browsing, Modifying, Deleting, or Adding Control Records

To browse, modify, or add control records, at the SNA : Control File Administration menu, do this:

Step 1. Enter **M** at the ===> prompt. The NEWS : Category Selection list is displayed.

> **Note**
>
> If you enter **L** on the SNA : Control File Administration menu, then you can only browse records from the selection list of records for a category.

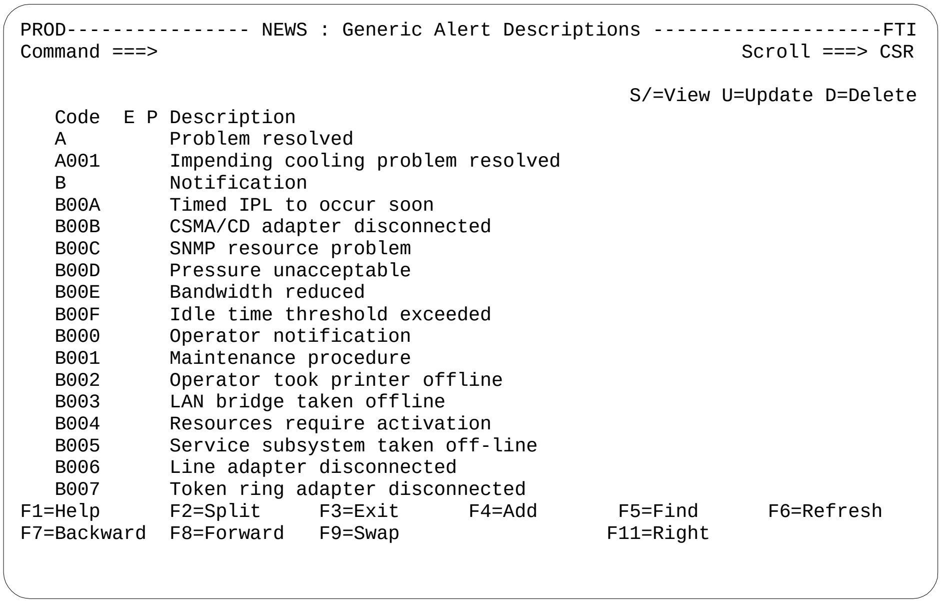
Figure 15-14.NEWS : Category Selection Panel



```
PROD-------------------- NEWS : Category Selection  ------------------------FTI
Command ===>                                                    Scroll ===> CSR

                                                                       S/=Select
    Cat Description
    001  Product-Set Identification
    002  Block Number Identification
    003  Record to Process-Id Conversion
    004  Process-Id Definitions
    009  Event Filtering by Event-ID Codes
    010  Event Descriptions and Probable Causes
    011  Basic Alert General Causes
    012  Basic Alert Specific Causes
    014  Basic Alert Text by Detail Text Reference Code
    015  Basic Alert Text by Description Code
    016  Basic Alert Text by User Action Code
    020  Generic Alert Descriptions
    021  Generic Alert Probable Causes
    022  Generic Alert User Causes
    023  Generic Alert Install Causes
    024  Generic Alert Failure Causes
    025  Generic Alert Qualified Message Data
 F1=Help      F2=Split     F3=Exit      F4=Return    F5=Find      F6=Refresh
 F7=Backward  F8=Forward   F9=Swap                  F11=Right
```

Step 2. To select a category from the list, enter **S** next to the category you want. A selection list of records for that category is displayed. Figure 15-15 is an example.

Figure 15-15.NEWS : Generic Alert Descriptions Panel



```
PROD---------------- NEWS : Generic Alert Descriptions --------------------FTI
Command ===>                                                     Scroll ===> CSR

                                                     S/=View U=Update D=Delete
   Code  E P Description
   A         Problem resolved
   A001      Impending cooling problem resolved
   B         Notification
   B00A      Timed IPL to occur soon
   B00B      CSMA/CD adapter disconnected
   B00C      SNMP resource problem
   B00D      Pressure unacceptable
   B00E      Bandwidth reduced
   B00F      Idle time threshold exceeded
   B000      Operator notification
   B001      Maintenance procedure
   B002      Operator took printer offline
   B003      LAN bridge taken offline
   B004      Resources require activation
   B005      Service subsystem taken off-line
   B006      Line adapter disconnected
   B007      Token ring adapter disconnected
F1=Help      F2=Split     F3=Exit      F4=Add       F5=Find      F6=Refresh
F7=Backward  F8=Forward   F9=Swap                  F11=Right
```

From this list you can select records to browse, modify, update, or delete. You can also add new records.

Browsing Control Records

To browse a record from the selection list of records for a category, enter **S** next to it. The selected record is displayed, in Browse mode.

Modifying Control Records

Step 1. To modify a record from the selection list of records for a category, enter **U** next to it. The selected record is displayed, in Update mode.

Step 2. Modify the record as required. For further details of how to modify each type of record, press F1 (Help).

Step 3. Press F3 to file your changes.

Deleting Control Records

Step 1. To delete a record from the selection list of records for a category, enter **D** next to it. A message is displayed, asking you to confirm your delete request.

Step 2. Press Enter to confirm your delete request or F12 to cancel the request.

Step 1. To add a new record to the selection list of records for a category, press F4 (Add). A panel for record details of that type is displayed, in Add mode.

Step 2. Enter details of the new record. For further details of how to define each type of record, press F1 (Help).

Step 3. Press F3 to file your changes.

Note

Control records are stored in the Network Services Control File (NSCNTL), which is normally shared between multiple NetMaster for SNA regions. Before you can update records, the file must be open for update and limited to one region. To change the NSCNTL file options, see the section, *Implementing the Network Services Control File (NSCNTL)*, on page 6-3.

Maintaining Resource Alias Names

NEWS provides VTAM with alias name translation services for those levels of VTAM that request this function. Alias names are used to differentiate between same name resources in interconnected networks.

Note

Alias name translation is not necessary if there are no resource name clashes when sessions are being established between interconnected networks. A name clash occurs if a resource name in one network is also defined in the other network.

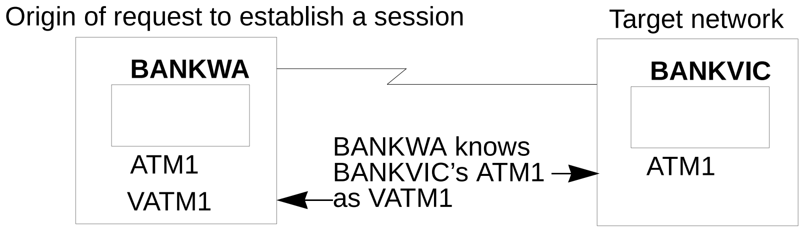
About Alias Name Translation

You can maintain the translation definitions by using the DEFALIAS REPALIAS, and DELALIAS commands.

You do not need to restart Management Services after changing or adding definitions. However, you may not be able to immediately use the new definitions for session establishment.

In Figure 15-16, *Example of Alias Name Translation*, the BANKWA network defined the alias name VATM1 to the resource ATM1 existing in the BANKVIC network because the name ATM1 was already assigned in the BANKWA network.

Figure 15-16.Example of Alias Name Translation


Origin of request to establish a session
Target network
BANKWA
BANKVIC
ATM1
VATM1
BANKWA knows BANKVIC's ATM1 as VATM1
ATM1

When multiple SNA networks are connected using a gateway function called SNA Network Interconnections (SNI), each network is known by a unique network identifier but otherwise retains its individual SNA characteristics.

During cross-network sessions in such an environment, resources that exist in a particular network may need to be known by an alias name in other networks.

Consequently, the process of establishing a cross-network session may require alias names in one network to be translated to the real resource names in another network. The Alias Name Translation Facility provides this service.

The facility can meet VTAM requests for translation both from the alias name to the real name, and from real name to alias name.

Session establishment also requires the use of Class-of-Service (COS) and Logmode names. Such a name may be defined in one network but unknown in another network. However, that other network may have an equivalent definition of the name. The Alias Name Translation Facility can be used to resolve the name difference between the networks.

Example of Resource Alias Name Translation

An LU, named X, in network A needs to connect to an application in network B. However, there is already a resource named X in network B. For a session to be established, an alias name for use in network B needs to be provided for the resource X in network A.

Example of Class-of-Service or Logmode Alias Names Translation

A file transfer application in network A always uses logmode X, and needs to connect to an application in another network, B. If the logmode X does not exist in network B, but an equivalent logmode exists, then the Alias Name Translation Facility can be used to assign the equivalent logmode in network B.

Displaying Alias Name Definitions

By using the SHOW DEFALIAS command, you can display one or more alias definitions used by the Alias Name Translation Facility of NEWS. By default, you can have an authority level of 0 to display alias name definitions.

For a full explanation and examples of the SHOW DEFALIAS command and details of operands, see the *Management Services Command Reference* manual.

Example of Displaying Alias Names

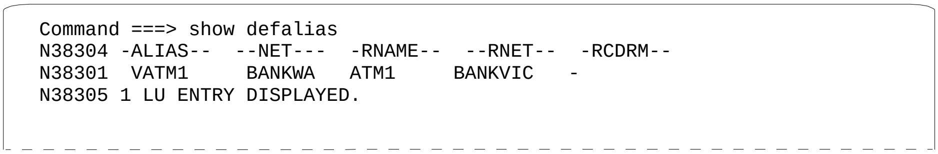
Figure 15-16, *Example of Alias Name Translation*, shows the alias name VATM1 defined in BANKWA for the real name ATM1 existing in BANKVIC.

To display the defined alias name, enter at a command line:

```
SHOW DEFALIAS
```

All defined alias names are displayed in a list (see Figure 15-17*)*.

Figure 15-17.SHOW DEFALIAS Results



```
Command ===> show defalias
N38304 -ALIAS--  --NET---  -RNAME--  --RNET--  -RCDRM--
N38301  VATM1     BANKWA   ATM1     BANKVIC   -
N38305 1 LU ENTRY DISPLAYED.
```

The definitions of the results are explained below.

ALIAS
: The alias name.

NET
: The name of the network in which the alias resource name is to be known, and the origin of the translation request.

RNAME
: The real name of the resource as it is known in the target network.

RNET
: The network identifier for the target network in which the real resource name can be used.

RCDRM
: (For LUs only) The CDRM that owns the LU.

Defining Alias Names

By using the DEFALIAS command, you can add an alias name to NEWS for use by the Alias Name Translation Facility. By default, you must have an authority level of 4 to add an alias name.

For a full explanation and examples of the DEFALIAS command and details of operands, see the *Management Services Command Reference* manual.

Note

The addition of definitions is normally a function of the INIT procedure processing.

Example of Defining an Alias Name

Figure 15-16, *Example of Alias Name Translation*, shows the alias name VATM1 defined in BANKWA for the real name ATM1 existing in BANKVIC.

To define the alias name of VATM1 as an LU in BANKWA, at a command line, enter:

```
DEFALIAS NAME=VATM1 NET=BANKWA
        RNAME=ATM1 RNET=BANKVIC
```

A message confirming the definition is displayed.

To see the result of the definition, at a command line, enter:

```
SHOW DEFALIAS NAME=VATM1
```

For more information about alias name definitions, see the section, *Defining Alias Names*, on page 15-29 in this chapter.

Defining Generic Names

You can reduce the number of DEFALIAS commands used and simplify subsequent modifications by defining generic alias names and network names.

You can also override the generic definitions by one or more specific conditions.

For a full explanation and examples of the command and details of operands, see the *Management Services Command Reference* manual.

Defining Generic Alias Names

You can define a generic alias name and real name pair when you want to map a range of similarly named resources (for example, MAIVF001 to MAIVF999) to some other range (for example, AMF001 to AMF999) in the target network.

By generically defining only the two prefix strings (that is, MAIVF and AMF), the Alias Name Translation Facility can carry the trailing suffix during the translation (that is, it translates MAIVF034 to AMF034).

Defining Generic Network Names

You can generically define the networks in which an alias name is known. By using a totally generic network name (that is, a name that any network name matches), in a single DEFALIAS command, you can define an alias to exist in all networks.

When a network name has been generically defined, any network name that matches the generic network name will contain the alias resource name defined to that network.

Replacing Alias Names

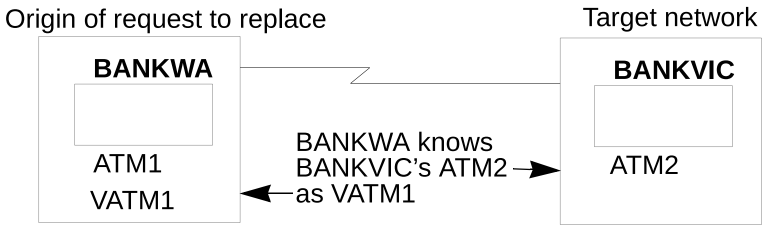
By using the REPALIAS command, you can replace the real name and network defined for an existing alias name and network combination. You must have an authority level of 4 to replace an alias name.

For a full explanation and examples of the REPALIAS command and details of operands, see the *Management Services Command Reference* manual.

Example of Replacing an Alias Name

Figure 15-18 shows the real name defined for the alias name VATM1 in BANKWA replaced with ATM2 existing in the BANKVIC network.

Figure 15-18.Replacing an Alias Name



To replace the real name defined for VATM1 in BANKWA, at a command line, enter:

```
REPALIAS NAME=VATM1 NET=BANKWA
        RNAME=ATM2 RNET=BANKVIC
```

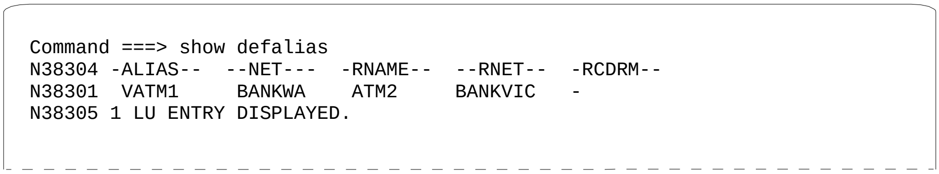
A message confirming the replacement is displayed.

To see the result of the replaced real name, enter at a command line:

```
SHOW DEFALIAS NAME=VATM1
```

The results, shown in Figure 15-19, are displayed.

Figure 15-19.Replaced Alias Name Results



```
Command ===> show defalias
N38304 -ALIAS--  --NET---  -RNAME--  --RNET--  -RCDRM--
N38301  VATM1     BANKWA    ATM2     BANKVIC   -
N38305 1 LU ENTRY DISPLAYED.
```

The definitions of the results are explained below.

ALIAS and NET

The resource name and network ID that identify the real name definition being replaced.

RNAME

The real name that replaces the previously defined real name.

RNET

The real network ID that replaces the previously defined real network ID.

Deleting Alias Names

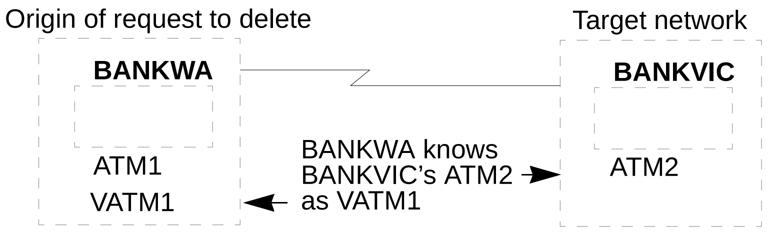
By using the DELALIAS command you can delete the alias name defined for a real resource in a target network. By default, you must have an authority level of 4 to replace an alias name.

For a full explanation and examples of the DELALIAS command and details of operands, see the *Management Services Command Referencee* manual.

Example of Deleting an Alias Name

Figure 15-20 shows the alias name of VATM1 in BANKWA defined for ATM2 in BANKVIC.

Figure 15-20.Deleting an Alias Name



To delete the alias name VATM1, at a command line, enter:

```
DELALIAS NAME=VATM1 NET=BANKWA
```

A message confirming the deletion is displayed.

Testing Alias Names Translation

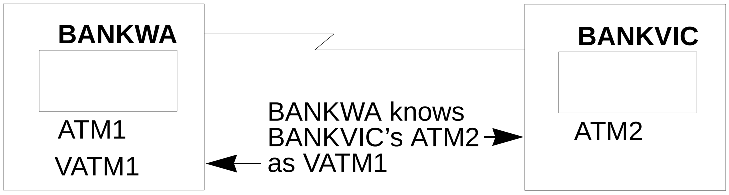
By using the XLATE command, you can test alias name translation. This command allows you to see the translated name that the Alias Name Translation Facility returns to VTAM when requested to perform translation. By default, you must have an authority level of 1 to test alias name translation.

For a full explanation and examples of the XLATE command and details of operands, see the *Management Services Command Reference* manual.

Examples of Testing Translation

Figure 15-21 shows the testing performed to determine that real name ATM2 existing in network BANKVIC is translated to the alias name VATM1 for the target network of BANKWA.

Figure 15-21.Testing Alias Name Translation



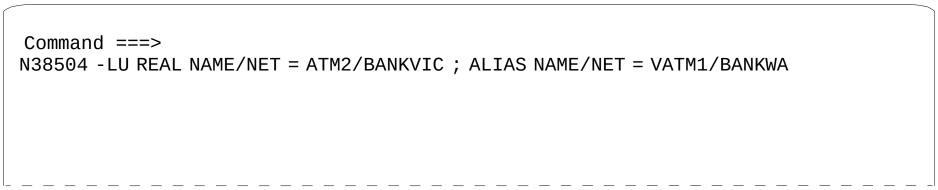
Example 1

To test that real name ATM2 existing in network BANKVIC is translated for the target network of BANKWA for the alias name of VATM1, at a command line, enter:

```
XLATE NAME=ATM2 NET=BANKVIC TARGNET=BANKWA REAL
```

The results, shown in Figure 15-22, are displayed.

Figure 15-22.Alias Name Testing, Example 1



```
 Command ===>
N38504 -LU REAL NAME/NET = ATM2/BANKVIC ; ALIAS NAME/NET = VATM1/BANKWA
```

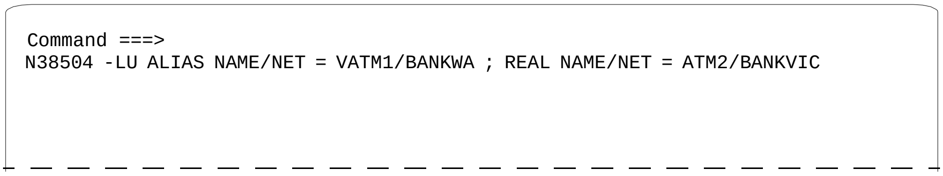
Example 2

To test that the alias name VATM1 in network BANKWA is translated to the real name ATM2 for the target network BANKVIC, at a command line, enter:

```
XLATE NAME=VATM1 NET=BANKWA TARGNET=BANKVIC
```

The results, shown in Figure 15-23, are displayed.

Figure 15-23.Alias Name Testing, Example 2



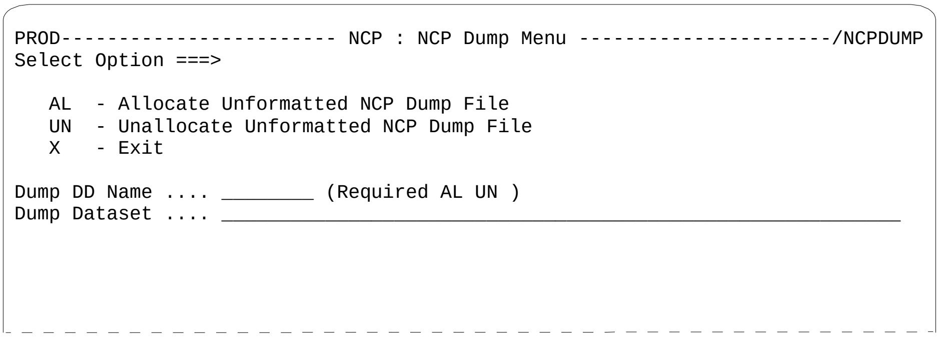
```
 Command ===>
N38504 -LU ALIAS NAME/NET = VATM1/BANKWA ; REAL NAME/NET = ATM2/BANKVIC
```

Allocating NCP Unformatted Dumps

To use an NCP unformatted (raw) dump, you need to allocate it to NCPView, so that NCPView can access information in the dump as though it is a real NCP.

To access the NCP Dump Menu, enter **/NCPDUMP** at a ===> prompt.

Figure 15-24.NCP : NCP Dump Menu



```
PROD------------------------ NCP : NCP Dump Menu ----------------------/NCPDUMP
Select Option ===>

   AL  - Allocate Unformatted NCP Dump File
   UN  - Unallocate Unformatted NCP Dump File
   X   - Exit

Dump DD Name .... ________ (Required AL UN )
Dump Dataset .... ___________________________________________________________
```

From this menu, you can allocate (option AL) or unallocate (option UN) an unformatted NCP Dump file. For further information, press F1 (Help).

Note

The DD Name that is specified must not conflict with DDs already allocated to the NetMaster system and also cannot conflict with an existing NCP name.

When the process is complete, the following message is displayed:

```
ZNC0702 FUNCTION COMPLETED SUCCESSFULLY
```

Example

If a dump file was allocated using the option **AL** on the NCPView Control Functions menu with a DD name of PRODDUMP specified, then the NCP selection list includes an NCP with the name PRODDUMP with all the information that a real NCP has displayed. This line on the list is displayed in blue to distinguish it from real NCPs.

Expected Unformatted Dump File Characteristics

The first record in the unformatted dump file is a control record. The device type that produced the dump is indicated in the first word of the control record.

The format of the first word is XXXXXXTT. The following are possible values of the TT byte:

- X‘00’ indicates a 3705 dump (not supported by NCPView)
- X‘01’ indicates a 3725/3720
- X‘02’ indicates a 3745

In a valid 3725/3720/3745 NCP dump, the actual NCP storage begins in the second record. The first word of the second record must contain X‘714C01AA’.

The LRECL of the dump must be equal to 512 or 2048.

Estimating Storage Requirements for Processing NCP Dumps Using NCPView

When a dump is accessed by a user, only the required amount of storage is read into memory. For example, if you browsed storage that was in the middle of the dump, then only half of the dump would be read into memory. If the dump is not accessed for 30 minutes, then the dump stored in memory is released, freeing memory. Therefore, further access to the dump would cause the dump to be read into memory again.

Most dumps are 4 or 8 Mb in size, although they can be up to 16 MB in size.

When considering how much virtual storage a dump may consume, the general rule is:

storage_required = size _of _dump + 300K

Ensure that the size of your region is set to an appropriate value.

Note

All storage is above the 16 MB line.

Integrating NCS with a Configuration Management Database

The default variables present in the initialization procedure, $NSINIT, and the distributed NCL exit procedure, $NCCNFG, enable NCS to retrieve data from a SOLVE:Configuration database.

If you have SOLVE:Configuration in your product region, NCS automatically uses the SOLVE:Configuration database to display configuration information.

If you do not have SOLVE:Configuration in your product region, you have two options:

- To tailor the $NSINIT procedure to integrate NCS with the INFO/MASTER product, if you have it
- To tailor the $NCCNFG procedure, if configuration data is to be retrieved from a source other than the SOLVE:Configuration or the INFO/MASTER product

About Exit $NCCNFG

The $NCCNFG procedure uses global NCL variables to determine the type of configuration database to access. For more information on this procedure, see the comments contained in the procedure.

When $NCCNFG Is Used

$NCCNFG is an NCL procedure that is called by NCS under the following conditions:

- When the F10 (Configuration) key is pressed to retrieve node descriptions for each of the nodes in a selection list
- When the selection option C is entered beside a node, to request the presentation of a panel or a series of panels displaying (configuration) data that is associated with the selected node

This procedure uses global NCL variables to determine the type of configuration database to access. For details, see the comments provided in the $NCCNFG procedures and in the $NSINIT procedure.

Integrating with INFO/MASTER

If your region is configured for INFO/MASTER and not for SOLVE:Configuration, then configuration data can be retrieved from the alternative source identified in the $NSINIT procedure; that is, a user configuration management system written using the INFO/MASTER product. (See the *INFO/MASTER Programmer's Reference, Planning, and Installation* manual for details on setting up a configuration management database.)

When obtaining data from an INFO/MASTER system, the $NCCNFG procedure uses four global variables to store the requisite identifiers. The default values shown in the following table are assigned in the distributed $NSINIT procedure. These values are those required for integration with the sample configuration management system that is distributed with the INFO/MASTER product:

Global Variable	Description	Default Value
&&000$NSCCSYS	INFO/MASTER system name	SAMPLE
&&000$NCSCCAT	INFO/MASTER category name	CONF
&&000$NCSCNDB	INFO/MASTER NDB name	IMNDB
&&000$NCSCPRC	INFO/MASTER scan procedure name	$CFSCAN

To integrate NCS with your installation of INFO/MASTER, do this:

Step 1. Access the NetMaster for SNA INIT procedure.

Step 2. For the variable &&000$NSCCSYS, specify your INFO/MASTER system name.

Step 3. For the variable &&000$NCSCCAT, specify your INFO/MASTER category name.

Step 4. For the variable &&000$NCSCPRC, specify your INFO/MASTER scan procedure name.

Step 5. Ensure that the &&000$NCSCNDB variable is set.

Note

The NDB name is set by calling procedure $IMSYINT after INFO/MASTER initialization; that is, after the Configuration Management system has been defined to INFO/MASTER.

Integrating with Any Other Source

If your region is not configured for either SOLVE:Configuration or INFO/MASTER, and so has to retrieve configuration information from another source, then you must tailor the distributed exit $NCCNFG.

For information on how to tailor this exit procedure, see the comments supplied in the procedure.

Creating User Alerts

Alerts are used to report user-defined events or to test processing within the structure of the NEWS system. Alerts appear as events and possibly attention messages on the NEWS Real-time Attentions Display

Alerts are generated by NCL procedures using the various string handling functions within NCL. All alerts must consist of valid expanded-hexadecimal characters. No restrictions are placed on the format of the alert record, although this capability would generally be used to produce user alerts with a standard CNM record format.

The completed alert is delivered directly to the CNMPROC in the nominated NEWS system using the &CNMALERT verb. The receiving system treats the alert as an unsolicited CNM record. The record is processed in the same manner as a record received from VTAM through the CNM interface.

The Inter-System Routing (ISR) facilities in NEWS are used to deliver alerts to remote NEWS systems. The destination system can be specified using the Link name, SSCP name or Domain name on the &CNMALERT statement. For more information on the &CNMALERT verb, see the *Network Control Language Reference*.

Uses for Alerts

Alerts can be used to test the CNM record arrival, and display processing paths of the receiving system. This permits the testing of record support which may not otherwise be possible until a CNM record arrives through the CNM interface. If the processing path for the record is incomplete or incorrect, or if a processing procedure fails, then a valuable record may be lost.

Sending Alerts from NCL Procedures

All alerts you create using the NEWS Create an Alert facility are constructed and sent using the $NWALERT utility procedure.

NEWS provides facilities for NCL procedures to create alerts and direct them to a CNMPROC running in a local or remote system. The alerts created by the &CNMALERT verb are processed by CNMPROC in the same manner as unsolicited records which have been received from VTAM through the CNM interface.

The alerts created by NCL procedures are always queued directly to the CNMPROC running on the nominated system. The alerts are not sent to VTAM and do not solicit a reply, but only pass information containing a user-defined event to CNMPROC.

Using the $NWALERT Procedure

$NWALERT is a utility procedure which provides support for other NCL procedures and assists in the formatting and sending of NMVT alerts. This procedure uses expanded-hexadecimal data from variables shared by the calling procedure to create and send the alerts. Any NCL procedure in the system may use this utility to produce an alert.

The alerts are constructed from information passed to $NWALERT by the calling procedure in a set of NCL variables which are shared by the calling procedure using the &CONTROL SHRVARS option. These variables contain codes which are to be placed in the alert record. All data passed to the $NWALERT procedure is validated before being placed in the relevant subvectors. If any supplied data fails validation, the alert is not constructed and an error message is returned to the calling procedure in the &SYSMSG variable to indicate the nature of the error. The comment block at the start of the $NWALERT procedure contains a description of the data required to produce an alert.

An alert can be sent to a remote NEWS system for processing. You can supply a link name or SSCP name to indicate that the alert is to be delivered to a remote NEWS system for processing. The $NWALERT procedure provides support to allow for the use of the Inter-System Routing (ISR) facilities in NEWS to direct the alert to a remote system. If the local system is specified, then ISR is not used and the alert is delivered to the local system for processing. For more information on this facility, see the &CNMALERT verb description in the *Network Control Language Reference* manual.

The alert is sent to the targeted NEWS system using the &CNMALERT verb. This verb queues the NMVT RU (embedded in a Deliver RU) to the CNMPROC procedure in the nominated system for processing. Delivery to the nominated system might not be possible. For example, the nominated CNMPROC may not be active or the selected system may not be available. If this is the case, then an error message is placed in the &SYSMSG variable to explain the reason for the failure. If the error message contains a standard message, then the online message help provides a more detailed explanation of the error.

If the alert is created and successfully delivered to the requested system, then a return code of 0 in the &RETCODE variable and a message confirming the successful completion of the operation in the &SYSMSG variable is returned to the calling procedure. If an error has been detected whilst the alert was being created or sent, the &RETCODE variable is set to a non-zero value and the &SYSMSG variable contains an error message. Both of these variables are returned to the calling procedure.

Creating Alerts

The NEWS : Create an Alert menu allows users to create the following NMVT alerts:

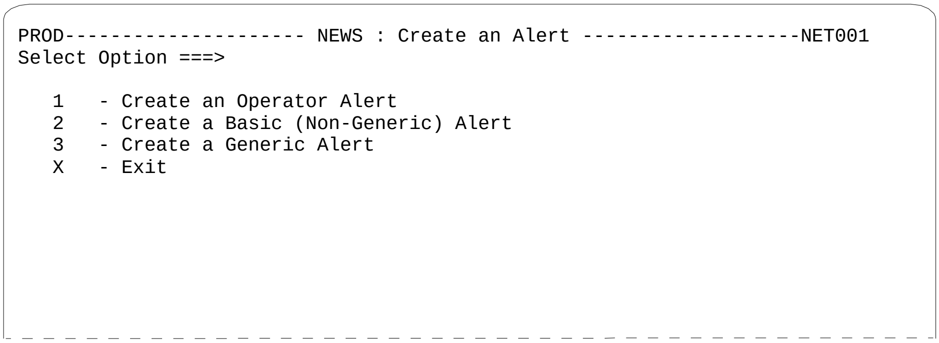
- Operator Alerts
- Non-generic (Basic) Alerts
- Generic Alerts

Before you use any of the options from the Create an Alert menu, you need to understand the NMVT alert structure. For this information, see the section, *Sending Alerts from NCL Procedures*, on page 15-40 in this chapter.

Accessing the NEWS Create an Alert Menu

To access the NEWS : Create an Alert Menu, enter **/SNADIAG.CA** at a ===> prompt.

Figure 15-25.NEWS : Create an Alert Menu



```
PROD--------------------- NEWS : Create an Alert -------------------NET001
Select Option ===>

   1   - Create an Operator Alert
   2   - Create a Basic (Non-Generic) Alert
   3   - Create a Generic Alert
   X   - Exit
```

To create an alert, enter the option number at the ===> prompt.

Creating an Operator Alert

You can produce operator alerts in the form of text messages to send to network operators.

To create an operator alert, enter **1** at the ===> prompt on the NEWS : Create an Alert Menu. The NEWS : Create an Operator Alert panel is displayed.

Complete the fields on the panel as follows:

1. In the Text Message field, type a maximum of 10 lines of text, each of 60 characters.

   The text is entirely free-form and can contain any information required by the operator.

2. In the Node Name field, if you want to change the default, then type the name of a resource in the Node Name field.

   By default, the user ID of the operator creating the alert is used as the name of the resource sending the alert. The receiving NEWS system logs the record in the NEWS database under the name of the resource which sent the alert.

3. If remote routing is required, do this:

   a. In the Link Name field, type a link name to send the alert to the associated remote NetMaster for SNA system.

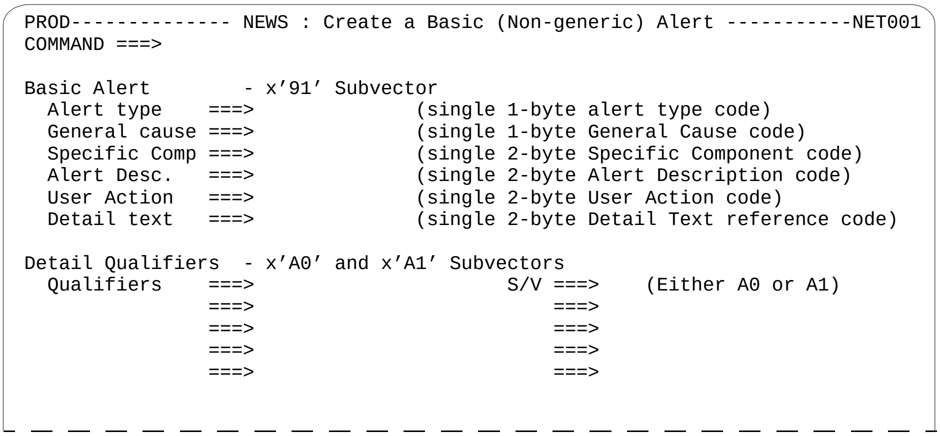
   b. In the SSCP Name field, type an SSCP name to send the alert to the associated remote NetMaster for SNA system.

Creating a Non-generic (Basic) Alert

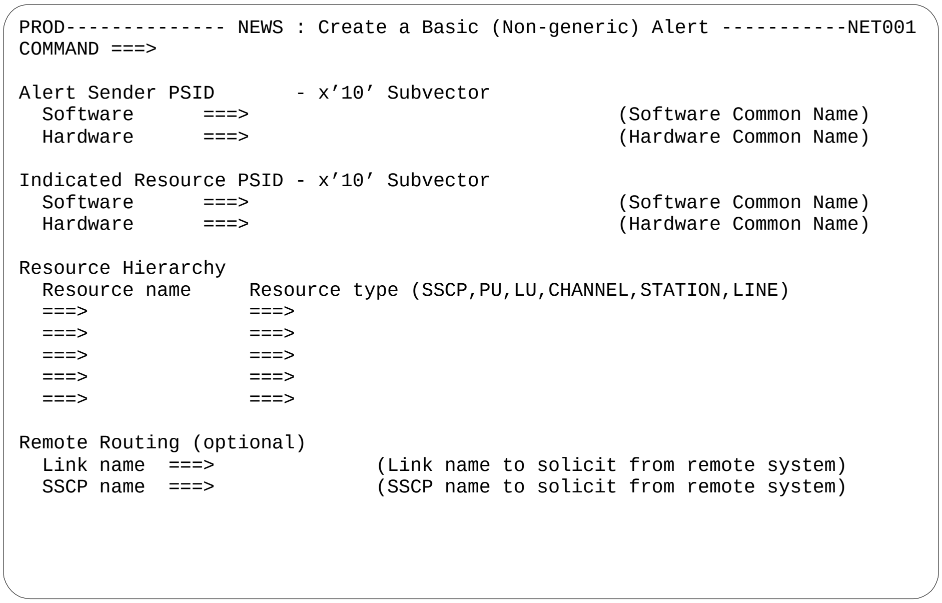
You can create a basic NMVT Alert to report user-defined events, or to test the existing CNM processing path for any type of alert. The alert is queued to the targeted CNMPROC (on a local or remote system) for processing.

To create a non-generic (basic) alert, enter **2** at the ===> prompt on the NEWS : Create an Alert Menu. The first NEWS : Create a Basic (Non-generic) Alert panel is displayed. Press F8 key to access the next panel. The panels are shown in Figure 15-26.

Figure 15-26.NEWS : Create a Basic (Non-generic) Alert Panel



```
PROD-------------- NEWS : Create a Basic (Non-generic) Alert -----------NET001
COMMAND ===>

Basic Alert        - x’91’ Subvector
  Alert type    ===>              (single 1-byte alert type code)
  General cause ===>              (single 1-byte General Cause code)
  Specific Comp ===>              (single 2-byte Specific Component code)
  Alert Desc.   ===>              (single 2-byte Alert Description code)
  User Action   ===>              (single 2-byte User Action code)
  Detail text   ===>              (single 2-byte Detail Text reference code)

Detail Qualifiers  - x’A0’ and x’A1’ Subvectors
  Qualifiers    ===>                      S/V ===>    (Either A0 or A1)
                ===>                          ===>
                ===>                          ===>
                ===>                          ===>
                ===>                          ===>
```



```
PROD-------------- NEWS : Create a Basic (Non-generic) Alert -----------NET001
COMMAND ===>

Alert Sender PSID       - x’10’ Subvector
  Software      ===>                                (Software Common Name)
  Hardware      ===>                                (Hardware Common Name)

Indicated Resource PSID - x’10’ Subvector
  Software      ===>                                (Software Common Name)
  Hardware      ===>                                (Hardware Common Name)

Resource Hierarchy
  Resource name     Resource type (SSCP,PU,LU,CHANNEL,STATION,LINE)
  ===>              ===>
  ===>              ===>
  ===>              ===>
  ===>              ===>
  ===>              ===>

Remote Routing (optional)
  Link name  ===>              (Link name to solicit from remote system)
  SSCP name  ===>              (SSCP name to solicit from remote system)
```

The panel provides fields the following alert information:

- Basic alert
- Detail qualifiers
- Alert Sender PSID
- Indicated Resource PSID
- Resource hierarchy
- Remote routing

See also the following sections, *Entering Alert Sender PSID and Indicated Resource PSID*, *Entering Resource Hierarchy Information*, and *Entering Remote Routing Information*, in this chapter.

Except for resource hierarchy and remote routing information, the alert information is built to form subvectors. Each subvector carries information that helps describe the alert condition. For a description of these subvectors, see the IBM publication, *SNA Formats and Protocols Reference*.

The Basic Alert can contain the subvectors shown in Table 15-1.

Table 15-1.Basic Alert Subvectors

Subvector	Description
X‘91’	Basic Alert subvector (required). This subvector describes the condition which led to the generation of the alert, the possible causes of the alert condition, the recommended user action, and may also supply a Detail Text Reference code to further describe the alert condition.
X‘A0’	Detail Qualifier Subvector (optional). This subvector can be present if a Detail Text Reference code was supplied in the X‘91’ subvector. It supplies a qualifier which is added to the Detail Text when it is displayed. The qualifier contained in this subvector is in character form and is not interpreted before being displayed.
X‘A1’	Detail Qualifier Subvector (optional). This subvector can be present if a Detail Text Reference code was supplied in the X‘91’ subvector. It supplies a qualifier which is added to the Detail Text when it is displayed. The qualifier contained in this subvector is in hexadecimal form and is translated into character format before being displayed.
X‘10’	Product Set ID Subvector (optional). This subvector describes a network resource. The alert can contain up to two of these subvectors. The first, if present, describes the resource sending the alert. This resource, called the Alert Sender, might be reporting an alert condition in another resource. If this is the case, a second Product Set ID subvector might be present, which describes the indicated resource. These resources are identified by their Common Hardware or Common Software name. For example, an IBM 3174 Control Unit would have a Common Hardware name of 3174.

To enter basic alert information for each subvector, do this at the NEWS : Create a Basic (Non-generic) Alert panel:

Step 1. From the *SNA Formats and Protocols Manual,* obtain the relevant reference code associated with the subvector.

Step 2. Type the code after the ===> field prompt.

Entering Detail Qualifiers

When you enter detail qualifiers, you specify how data for the detail text of an alert message is to be transmitted for display. The detail text describes in detail what condition caused the generation of the alert and is defined by the reference code you entered in the Detail Text field in the Basic Alert subvector.

To enter detail qualifiers at the NEWS : Create a Basic (Non-generic) Alert panel, do this at the Qualifiers and S/V ===> prompts:

1. To transmit the detail data in text characters, use the text contents of the subvector X‘91’ to type, in each Qualifiers field, the text characters for the detail data.

   Otherwise, to transmit the hexadecimal representation of the data, type the hexadecimal equivalent of the contents of the subvector X‘91’. The hexadecimal representation is converted to EBCDIC text before being displayed.

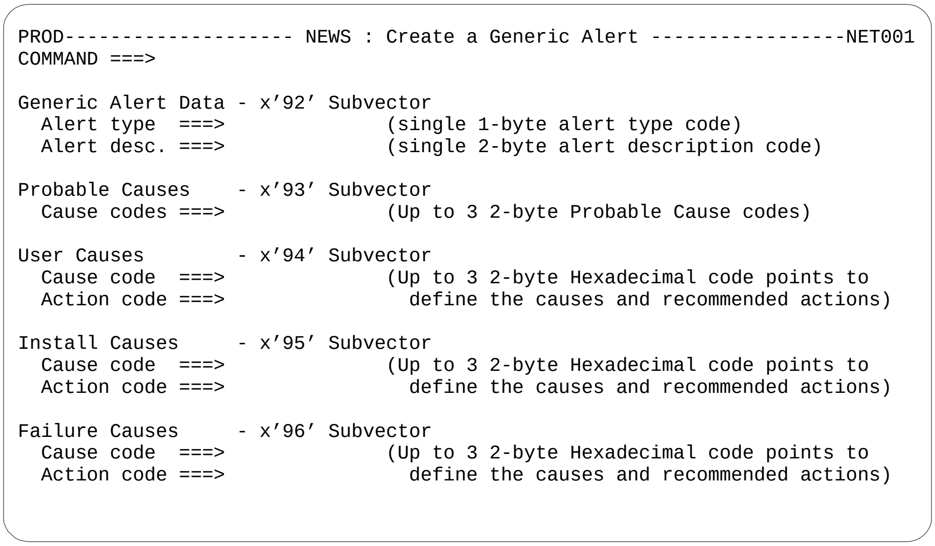
2. In each S/V field, type **A0** for text qualifiers, or **A1** for hexadecimal qualifiers.

Creating a Generic Alert

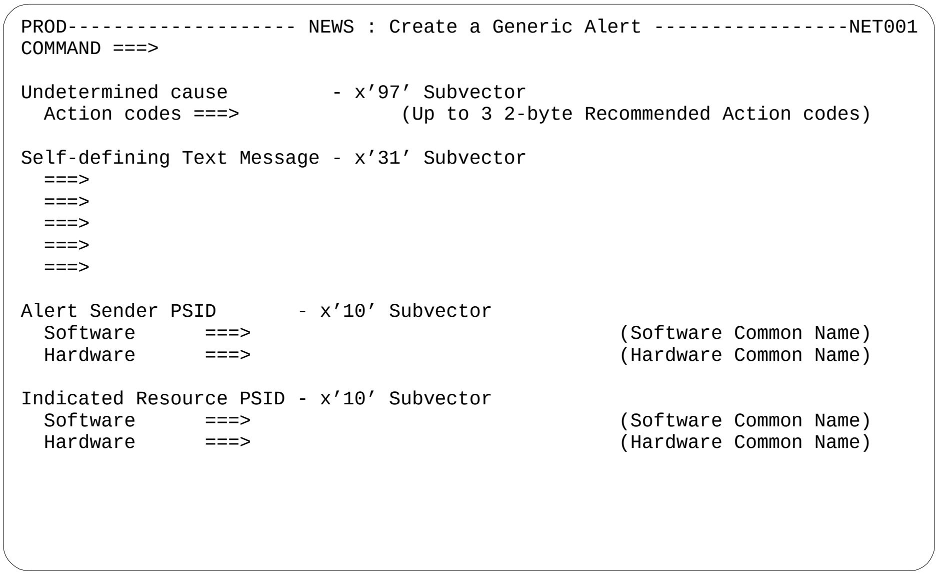
You can create Generic NMVT Alerts to report events in the network or to test the existing CNM processing path for such an alert. The alert is queued to the targeted CNMPROC (on a local or remote system) for processing.

To create a generic alert, ENTER **3** at the ===> prompt on the NEWS : Create an Alert menu. The first NEWS : Create a Generic Alert panel is displayed. Press the F8 key to access the next panels. The panels are shown in Figure 15-27.

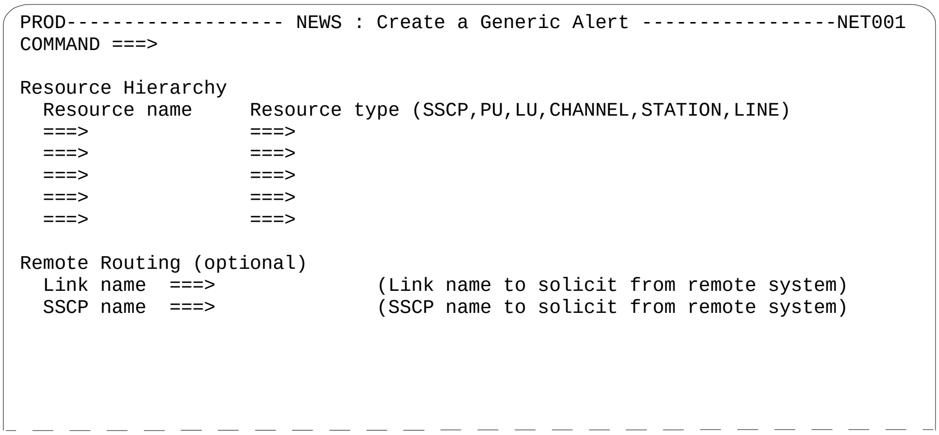
Figure 15-27.NEWS : Create a Generic Alert Panel



```
PROD-------------------- NEWS : Create a Generic Alert -----------------NET001
COMMAND ===>

Generic Alert Data - x’92’ Subvector
  Alert type  ===>             (single 1-byte alert type code)
  Alert desc. ===>             (single 2-byte alert description code)

Probable Causes    - x’93’ Subvector
  Cause codes ===>             (Up to 3 2-byte Probable Cause codes)

User Causes        - x’94’ Subvector
  Cause code  ===>             (Up to 3 2-byte Hexadecimal code points to
  Action code ===>               define the causes and recommended actions)

Install Causes     - x’95’ Subvector
  Cause code  ===>             (Up to 3 2-byte Hexadecimal code points to
  Action code ===>               define the causes and recommended actions)

Failure Causes     - x’96’ Subvector
  Cause code  ===>             (Up to 3 2-byte Hexadecimal code points to
  Action code ===>               define the causes and recommended actions)
```



```
PROD-------------------- NEWS : Create a Generic Alert -----------------NET001
COMMAND ===>

Undetermined cause         - x’97’ Subvector
  Action codes ===>              (Up to 3 2-byte Recommended Action codes)

Self-defining Text Message - x’31’ Subvector
  ===>
  ===>
  ===>
  ===>
  ===>

Alert Sender PSID       - x’10’ Subvector
  Software      ===>                                (Software Common Name)
  Hardware      ===>                                (Hardware Common Name)

Indicated Resource PSID - x’10’ Subvector
  Software      ===>                                (Software Common Name)
  Hardware      ===>                                (Hardware Common Name)
```



```
PROD------------------- NEWS : Create a Generic Alert -----------------NET001
COMMAND ===>

Resource Hierarchy
  Resource name     Resource type (SSCP,PU,LU,CHANNEL,STATION,LINE)
  ===>              ===>
  ===>              ===>
  ===>              ===>
  ===>              ===>
  ===>              ===>

Remote Routing (optional)
  Link name  ===>              (Link name to solicit from remote system)
  SSCP name  ===>              (SSCP name to solicit from remote system)
```

The panel provides fields for the following alert information:

- Generic alert data
- Probable causes
- User causes
- Install causes
- Failure causes
- Undetermined cause
- Self-defining text message
- Alert sender PSID
- Indicated resource PSID
- Resource hierarchy
- Remote routing

Except for resource hierarchy and remote routing information, the alert information is built to form subvectors. Each subvector carries information that helps describe the alert condition. For a description of these subvectors, see the IBM publication, the *SNA Formats and Protocols Manual*.

Entering Generic Alert Information

The Generic Alert can contain the subvectors shown in Table 15-2.

Table 15-2. Generic Alert Subvectors

Subvector	Description
X'92'	Generic Alert subvector (required). This subvector describes the severity of the condition which led to the generation of the alert, and gives a code which describes the alert condition.
X'93'	Probable Causes subvector (required). This subvector provides codes which describe the probable causes of the alert condition. A maximum of three Probable Cause codes can be entered.
X'94'	User Causes subvector (optional). This subvector provides codes which describe possible user-related causes of the alert condition, and supplies Recommended Action codes. A maximum of three User Cause codes and three Recommended Action codes can be entered.
X'95'	Install Causes subvector (optional). This subvector provides codes describing errors which might have been made during the installation of the resource which may have caused the alert condition, and supplies Recommended Action codes. A maximum of three Install Cause codes and three Recommended Action codes can be entered.
X'96'	Failure Causes subvector (optional). This subvector provides codes which describe possible device or software failures which may have caused the alert condition, and supplies Recommended Action codes. A maximum of three Failure Cause codes and three Recommended Action codes can be entered.
X'97'	Undetermined Cause subvector (optional). If the cause for the alert condition is not known or cannot be expressed in the previous cause code subvectors, this subvector must be included in the alert. It carries no Cause codes, but specifies Recommended Action codes to describe the action necessary. A maximum of three Recommended Action codes can be entered.

Table 15-2.Generic Alert Subvectors

Subvector	Description
X‘31’	Self-Defining Text Message subvector (optional). Each X‘31’ subvector carries text which can help with further diagnosis of the condition leading to the generation of the alert.
X‘10’	Product Set ID Subvector (optional). This subvector describes a network resource. The alert can contain up to two of these subvectors. The first, if present, describes the resource sending the alert. This resource, called the Alert sender, might be reporting an alert condition in another resource. If this is the case, a second Product Set ID subvector may be present, which describes the indicated resource. These resources are identified by their Common Hardware or Common Software name. For example, an IBM 3174 Control Unit would have a Common Hardware name of 3174.

To enter generic alert information for each subvector, do this at the NEWS : Create a Generic Alert panel:

Step 1. From the *SNA Formats and Protocols Manual,* obtain the relevant code associated with the subvector.

Step 2. Type the code after the ===> field prompt.

Entering the Self-defining Text Message

The text message helps the sender explain the reason for the alert.

To enter the text message, at the NEWS : Create a Generic Alert panel, type up to five lines of text, each of 60 characters.

Entering Alert Sender PSID and Indicated Resource PSID

The subvector 10 identifies the sender of the alert and indicates the resource of concern (if the alert is being sent on behalf of another resource). The Product Set ID identifies devices and applications in the network.

To enter a PSID, do this at the NEWS : Create a Generic Alert panel:

Step 1. In the Software field, type a software product name.

Step 2. In the Hardware field, type a hardware product name.

Entering Resource Hierarchy Information

The Resource Hierarchy indicates the network hierarchy existing above this resource. Providing this information is optional.

To enter the resource hierarchy information, at the NEWS : Create a Generic Alert panel, do this:

Step 1. In descending order, type resource names.

Enter the hierarchy in descending order, so the resource immediately connected to the reported resource is the last in the hierarchy list.

Step 2. Type resource types. The valid resource types are SSCP, PU, LU, CHANNEL, STATION, and LINE.

Entering Remote Routing Information

The alert can be directed to a remote system for processing by using the NEWS ISR facilities.

To send the alert to a remote system, do this at the NEWS : Create a Generic Alert panel:

Step 1. In the Link Name field, type a link name to send the alert to the associated remote product system.

Step 2. In the SSCP Name field, type an SSCP name to send the alert to the associated remote product system.

Note

If no remote routing is requested, then the alert is directed to the local CNMPROC.

Part III

Reference

A

Security Exit Support Requirements

If you have installed a full security exit to replace the Management Services UAMS security component, then your exit must provide all processing associated with the retrieval and verification of user ID information normally performed by UAMS.

For information about the structure and method of operation of an installation-supplied full security exit, see the *Management Services Administrator Guide*. This appendix assumes that you are familiar with the procedure for installing a full security exit, as documented in that manual.

This appendix summarizes all structured fields that are specific to NetMaster for SNA. For detailed definitions of each field, see the *Structured Fields* appendix in the *Management Services Administrator Guide*.

Structured Field Descriptions

Table A-1 below lists the structured fields, the supported NetMaster for SNA component, and a brief description of the support.

Table A-1. List of Structured Fields

Structured Field	Component	Function
X‘0022’	-	Defines Network Management access.
X‘0026’	NEWS	Defines NEWS access privilege.
X‘0150’	NEWS	Defines NEWS reset privilege.
X‘0151’	NTS	Defines NTS access privilege.
X‘002D’	NCS	Defines NCS access privilege.
X‘0090’	NCPView	Defines NCPView access privilege.

Note

A prerequisite for all other access privileges is Network Management access.

For detailed definitions of each field, see the *Structured Fields* appendix in the *Management Services Administrator Guide*.

B

About the CNM Interface

To fully use the facilities that NEWS provides, you may need some understanding of how the CNM interface operates.

This appendix introduces the CNM interface, presenting background information about the general flow of CNM Request Units (RUs) and sessions. Additional information is provided about the RUs currently used by the distributed NEWS system, although all RU types can be supported by the Network Control Language (NCL).

CNM Application

The CNM interface provides a means by which a suitably authorized VTAM application program (referred to here as the CNM application) can maintain a session with the System Services Control Point (SSCP) of the VTAM under which the application is executing. A session is established when the application successfully opens its VTAM ACB, enabling it to exchange data with the SSCP.

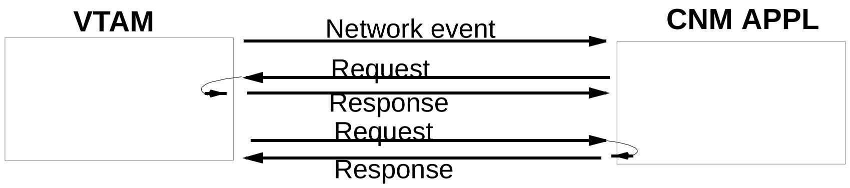
The CNM application can receive data from an SSCP in one of the following forms, as shown in Figure B-1:

- Unsolicited, as a result of some network event
- Solicited, as a reply to a previous request for data issued by the CNM application
- A solicitation, requesting that the CNM application send some reply data in response

The CNM application can send data to the SSCP for the following reasons:

- To solicit reply data from a network resource
- As a reply to a solicitation from the SSCP

Figure B-1. Data Exchange Between VTAM and CNM APPL



Network Services Request Units (NS RUs)

There are various types of NS RUs that can be received or sent by an application that is using the CNM interface:

- Deliver RUs – such as Record Formatted Maintenance Statistics (RECFMS) and Record Maintenance Statistics (RECMS) RUs – which deliver data to the CNM application
- Forward RUs – such as Request Maintenance Statistics (REQMS) RUs – which are sent by the CNM application to request data delivery
- Network Management Vector Transport (NMVT) RUs, which perform both delivery and request functions

If an NS RU is solicited, then reply data is always returned to the soliciting application. If the NS RU is unsolicited, then delivery is influenced by the contents of the VTAM CNM Routing Table. Other factors, such as the functional capabilities of the SSCP and the CNM application, also have a bearing on the nature of data exchanged across the CNM interface.

CNM Data from Network Resources

CNM data for a network resource is carried in the NS RU, which is embedded in either a Forward or a Deliver RU, for the SSCP-PU (or SSCP-LU) session. The NS RU defines exactly the type of service required, or the type of data received.

Any CNM requests not embedded in a Forward or Deliver RU are recognized by the CNM application as being sent to or from the SSCP itself.

Forward Request Units

When a CNM application requires an SSCP to perform a particular service, it sends a Forward RU to the SSCP with which the destination network resource is associated. Among other data, this RU contains the node name of the network resource to which the request applies.

Embedded within the Forward RU is the NS RU describing the service required. The following are the most common forms of embedded RUs:

- Request Maintenance Statistics (REQMS) RUs
- Network Management Vector Transport (NMVT) RUs

If a REQMS or NMVT is destined for a resource in the network, then the SSCP forwards the NS RU to the destination resource across an SSCP-PU session.

One or more NS RUs can be sent in reply by the network resource. These flow back to the originating SSCP, which in turn presents them, embedded in a Deliver RU, to the soliciting CNM application.

Deliver Request Units

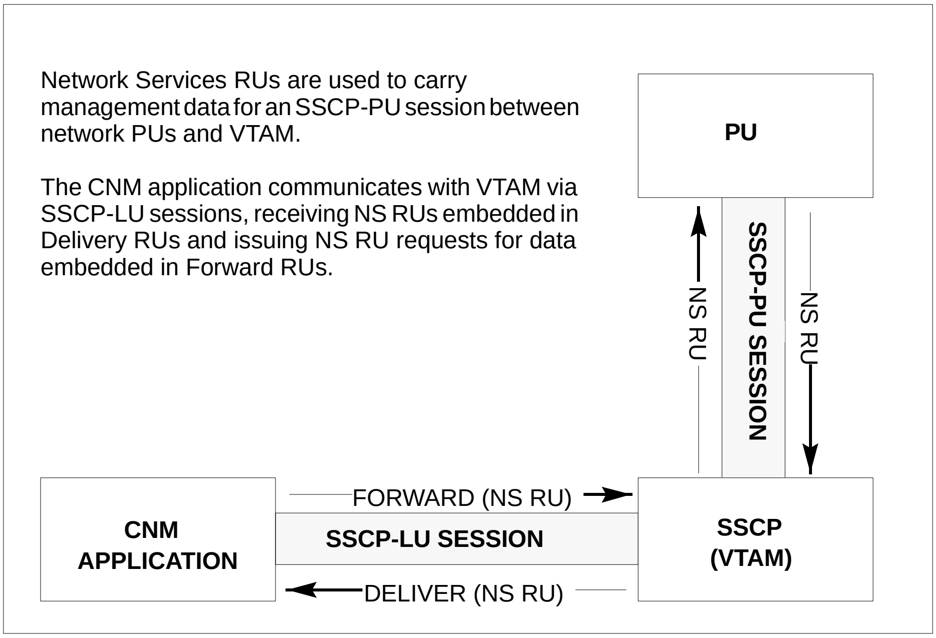
When an SSCP has some data to send to a CNM application on behalf of a network resource, it sends a Deliver RU to that application. The Deliver RU contains, among other things, the name of the network resource to which the data applies, and a resource hierarchy list.

Embedded within this RU is an NS RU which describes the type of data being made available. The most common forms of RUs are:

- Record Formatted Maintenance Statistics (RECFMS) RUs
- Record Maintenance Statistics (RECMS) RUs
- Network Management Vector Transport (NMVT) RUs

Figure B-2 illustrates the flow of NS RUs.

Figure B-2. Sessions Used by the Network Services RUs


Network Services RUs are used to carry management data for an SSCP-PU session between network PUs and VTAM.
The CNM application communicates with VTAM via SSCP-LU sessions, receiving NS RUs embedded in Delivery RUs and issuing NS RU requests for data embedded in Forward RUs.
PU
NS RU
SSCP-PU SESSION
NS RU
CNM APPLICATION
FORWARD (NS RU)
SSCP-LU SESSION
DELIVER (NS RU)
SSCP (VTAM)

NMVT NS Request Units

The Network Management Vector Transport RU provides a more generalized structure for carrying both requests and replies. It consists of one or more SNA MS major vectors that describe the type of network data contained within the request unit, each of which includes one or more Management Services sub-vectors.

An NMVT RU can be issued by the CNM application as a request for data.

Alternatively, an NMVT RU might be sent to the CNM application under the following circumstances:

- **Unsolicited**

  Certain devices generate unsolicited records in response to the occurrence of a local event. For example, some operator alerts are produced in the form of an NMVT RU.

- **Solicited**

  A reply to an NMVT RU might be sent in response to a request NMVT RU issued by the host application.

Generic and non-generic (Basic) NMVT alerts might also be sent by a device in the network to report an error or failure. The alert record contains data explaining the type of error or failure, the likely causes of the error or failure, and action that can be taken to remedy the situation.

Host CNM support for the 3x74 uses NMVT RUs to request Response Time Monitor data. An NMVT RU carrying RTM data might be sent by the 3x74 as an unsolicited record following a controller-detected event or as a reply to a solicitation request.

The following is the format of a Generic Alert NMVT:

Content	**Length**
CNM Header	8 bytes
NS-RU	8 bytes
Generic Alert Major Vector	Varying
Subvectors	Varying
Hierarchy information	Varying

REQMS NS Request Units

The REQMS NS RU (embedded in a Forward RU) is a means by which a CNM application can solicit data from a network resource. Six types of data that can be solicited are defined, although not all devices support all data types. Some devices support non-standard types and formats; therefore NEWS allows any format to be transmitted.

REQMS can only be sent to a resource that is owned by (that is, is in the domain of) the SSCP of the VTAM under which the CNM application is running, unless ISR is operating.

Defined REQMS Data Types

The six defined REQMS data types are described here:

REQMS Type 1
Solicits link test statistics from a Physical Unit (PU). PUs that support this function maintain details of the number of link test frames (from a VTAM Link-Level 2 test) received and the number responded to.

REQMS Type 2
Solicits summary counters from a PU. PUs that support this function maintain three categories of error counters: internal hardware errors, communications adapter errors, and negative responses.

REQMS Type 3
Solicits communications adapter errors from a PU. PUs that support this function maintain various categories of communications adapter error counters.

REQMS Type 4

Solicits PU/LU-dependent data from those PUs that support this type. The type of data sent varies according to the device type involved. For instance, REQMS type 4 RUs are sent to 3600/4700 subsystems to access the system monitor functions of that device type.

REQMS Type 5

Solicits EC-level data. PUs that support this function return data such as their microcode EC level, or part numbers installed. The reply format varies depending on the device type.

REQMS Type 6

Solicits link connection subsystem data. Used in conjunction with some modem types to retrieve link-related data.

RECFMS NS Request Units

The RECFMS RU is sent to a CNM application by an SSCP under two circumstances:

- As a reply to a previous REQMS request from the SSCP: when a network resource receives a REQMS, it formats a RECFMS in reply, and VTAM forwards it to the CNM application.
- As an unsolicited record: certain network resources can generate unsolicited RECFMS records under some circumstances.

Network resources always deliver RECFMS RUs to the SSCP that owns them (that is, the SSCP for the local domain).

Defined RECFMS Data Types

The types of RECFMS RUs are categorized in the same way as REQMS RUs. This means that a type 1 REQMS receives a type 1 RECFMS in reply (and so on), and that certain devices might generate a RECFMS of one of the types without being requested to do so. For instance, the 3600/4700 subsystem can generate RECFMS type 4 records to inform the host of a variety of conditions.

One additional RECFMS type exists, known as type 0. Because there is no matching REQMS type 0, the RECFMS type 0 is always unsolicited, and is classified as an alert message. Its content is not explicitly defined, so the exact data sent is device-dependent.

RECMS NS Request Units

NCPs in a network generate RECMS RUs under a variety of conditions, and there are a large number of RECMS types. Some of these types are described in the following examples:

- **Statistical**
  Each time certain counters for a particular node in the NCP reach a set threshold (that is, wrap), the NCP generates a statistical RECMS record, containing such data as traffic counts and temporary error counts. The frequency at which the counters wrap can be adjusted via NCP generation. The records are also forwarded whenever a node or the NCP itself is varied inactive (that is, the device is inoperative as far as the SSCP is concerned).
- **Error notification**
  A variety of error conditions cause the generation of RECMS records. These include permanent link and device failures, and temporary errors. The RECMS records provide an indication of the most likely cause of the error by including a snapshot of various NCP control blocks.
- **Software failure**
  Records can be generated as a result of NCP software failures or abends.
- **Hardware failure**
  Records can be generated as a result of communications controller hardware errors.

NCP delivers RECMS RUs to the SSCP(s) that own the resource to which they refer (that is, they are in the local domain of the SSCP).

SSCP Related CNM Requests

All the NS RUs discussed in the sections above are concerned with the transportation of network management requests between the host CNM application and network devices. As described, these requests are transported in one of the following ways:

- Embedded in a forward RU and passed to the SSCP for onward propagation to a network device
- Embedded in a deliver RU on arrival from a network device for delivery to the CNM application

Another class of requests also make use of the CNM interface. These requests contain data relating to the services of the SSCP directly and do not involve the redirection of RUs to other network devices. Such SSCP-related requests are either sourced from the SSCP and delivered to the CNM application, or are sourced from the CNM application for delivery to the SSCP. In either case, since no further propagation of the request is necessary, these requests are not embedded in a deliver RU (if sourced from the SSCP) or a forward RU (if sourced from the CNM application).

Some important SSCP-related CNM requests are described in the following sections.

Translate Inquiry and Reply Request Units

NEWS supports the unembedded *Translate-Inquiry* and *Translate-Reply* RUs. These request units are used for alias name translation by certain levels of VTAM.

The Translate-Inquiry RU is sent to NEWS from the SSCP and solicits a Translate-Reply RU in response. For more information on this subject, see *Maintaining Resource Alias Names*, on page 15-26.

CNM-RU

The CNM-RU is a control request unit used by the Network Tracking System (NTS) feature of NetMaster for SNA. It, too, is passed unembedded across the CNM interface and is ignored by NEWS.

How Records Are Processed

This section describes how the processing requirements of the record are determined using the Network Services Control File (NSCNTL), and how the requirements are met by the execution of the nominated NCL procedures.

Record Type Recognition

Each record received by NEWS via the CNM interface is either an RECMS record, an RECFMS record, or an NMVT record. The record type is further qualified as follows:

- For NMVT records, the MS major vector type
- For RECMS records, the recording mode
- For RECFMS records, the record type

Alerts received by NEWS from the APPN network arrive as SNA MSUs containing one or more major vectors. MSUs are processed as if they were NMVTs which also contain major vectors.

Processing Code Assignment

CNMPROC extracts the resource identifier from the record and uses the NSCNTL to identify the device from which the record was sourced. On this basis, the NSCNTL assigns two codes to the record to identify its processing requirements:

- A Resource ID (RID), used to group similar resource types
- An Event ID (EID), used to qualify later record processing

NMVT Records and MDS-MUs

NMVT records (and MSUs) normally contain a Product Set ID (PSID) sub-vector field that contains the hardware or software common name or machine type of the resource that sent the record. The PSID is used to obtain, from Category 001 (Product-Set Identification) of the NSCNTL table, a description of the resource and the associated RID and EID to be assigned to the record.

If no matching PSID exists in the NSCNTL table, the RID and EID are both set to UNKNOWN.

RECFMS Records

RECFMS records contain a block number field that identifies the type of resource that sent the CNM record. The block number is used to obtain from Category 002 (Block Number Identification) of the NSCNTL table a description of the resource, and the associated RID and EID to be assigned to the record. If no matching block number exists in the NSCNTL table, the RID and EID are both set to UNKNOWN.

RECMS Records

RECMS records have no resource identifier specified, and these codes can therefore not be assigned. Instead, RECMS records are assigned an Event ID during PID selection, as described in the next section.

Processing Path Selection

After the RID and EID have been selected, NEWS uses the NSCNTL to select a processing path for the record. This processing path describes the arrival and display processing requirements for the record, and is represented by a single control code called the PID. The PID is dependent on:

- The RU type
- The RID
- The record type (NMVT major vector type, RECMS recording mode, or RECFMS record type)

The PID is retrieved from Category 003—Record to PID Conversion—of the NSCNTL by CNMPROC. The PIDs (Category 004) then define the processing path for the records. This selection process allows common processing paths for different types of records. The selection process allows a PID to be assigned to any record type from any device.

Process Path Definitions

Process path definitions detail the processing requirements for the record. Included are the names of several NCL procedures that are used to perform NEWS record arrival processing functions. The procedures are classified into three groups:

- NEWS record arrival processing procedures, which are responsible for interpreting the contents of the record during processing by CNMPROC
- User intercept procedures, which provide further processing during processing by CNMPROC
- Display procedures which provide display formatting and presentation services. These procedures are executed to format and display data retrieved from the record.

These procedures are all optional. Where a procedure has been nominated and is enabled, it is executed by NEWS during record arrival processing.

Solicited CNM Record Processing

Solicited records can be processed by any NCL procedure, in the following way:

Step 1. Data is solicited from devices in the network by using the &CNMSEND verb.

Step 2. The replies to these solicitation requests are retrieved using the &CNMREAD verb.

Step 3. The READ= operand on the &CNMSEND verb is checked to determine whether the reply is to be processed by CNMPROC, the soliciting procedure, or both.

Step 4. If the reply is to be processed by the soliciting procedure, the $NWDSPLY procedure is usually executed. This procedure:

- Performs NEWS record arrival processing to determine the PID for the record
- Invokes display processing procedures nominated in the PID description to display the results

Step 5. If any logging is required, the reply should be directed to CNMPROC, because the $NWDSPLY procedure does not perform any SMF or NEWS database logging functions.

Unsolicited CNM Record Processing

All unsolicited CNM records received by NEWS are directed to CNMPROC for processing. CNMPROC does the following processing:

1. Determines the processing requirements for the record
2. Executes the procedures defined in the processing path for the record
3. Performs SMF and NEWS database logging, if required

References

The formats of individual records and RUs, and further details on the CNM interface appear in a number of IBM manuals, including:

- Communications Server manuals
- NCP/EP manuals
- SNA Architecture manuals

Various product-related manuals might also contain information about record formats that are peculiar to their device types.

C

About the Session Awareness (SAW) Interface

To fully use the facilities that NEWS provides, you may need some understanding of how the SAW interface operates. This appendix introduces the SAW interface.

Using NTS Classes

Within a given network, different session types need different types and amounts of data collected by NTS. It is also useful to be able to map this data onto the underlying resource hierarchy.

These objectives are achieved through use of the four types of NTS classes:

- SAW classes
- RTM classes
- Session classes
- Resource classes

When NTS receives a session start notification from VTAM, it determines, from the NTS class definitions, which options are to apply to the session. NTS extracts and stores these options, with other information about the session, for use in subsequent processing. This means that all NTS class definitions should be in place before collection of SAW data is started. Defining classes once SAW processing is active does not affect existing sessions, only new ones.

By default, if no classes are defined to NTS, SAW data only is collected for *all* sessions. Trace data can also be collected by operator request, but no accounting, RTM, or resource statistics data are collected.

SAW Classes

Each SAW class defined to NTS describes a set of processing options for all SAW information, including whether or not such information is required.

By default, NTS keeps in storage information concerning every session that is currently active. However, if this is not necessary, or is impracticable due to storage restrictions, then you must have a SAW class definition with the KEEP=NO option (using the DEFCLASS or REPCLASS command) to prevent NTS from collecting any data for sessions in that class. Since in this case no other SAW class options are meaningful, only one such SAW class definition should be necessary as many session classes can nominate the same SAW class definition to be used.

SAW class definitions specify the conditions under which all or any session data is to be logged to the NTS database at session end. For example, you may want to log session data if an error occurs that terminates the session, or perhaps whenever the operator collects trace data. Various SAW class options exist which cater for these and other requirements.

You can set the initial and final trace queue depths with the SAW class definition. This allows differing amounts of trace data to be kept for different sessions.

You can also use SAW classes to determine whether or not accounting information should be collected when NTS selective accounting is requested (that is, SYSPARMS NTSACCT=SELECTIVE is specified or defaulted).

RTM Classes

NTS allows you to collect RTM data for particular sessions. For NTS to be able to collect RTM information from network control units, you must define one or more RTM classes. In addition, it is necessary for the control units (which may be 3274s, 3174s or compatible devices) to have the required RTM hardware or microcode level support for the collection of RTM data and host-modifiable RTM definition configured.

Each RTM class specifies a set of up to four boundary values to be used for the collection of RTM data. These boundary values are ascending times in the range 0.1 seconds to 30 minutes. For sessions using the RTM class, these boundary values are set in the control unit for the duration of the session to capture the RTM data.

In addition, for each RTM class an *objective response time* and an *objective percentage* value are defined. These values can be used to represent a level of service so that you can compare the measured service level with that specified in a service agreement. You can also define *RTM definitions*, which are the response criteria that indicate what RTM data is kept.

The objective values are used in performance monitoring for network response times and can lead to attention message creation. For further information about these topics, see the section, *Response Time Monitor Data,* on page C-12.

The objective response time must correspond to one of the boundary values allocated in order for the objective percentage comparison to be accurate. It is suggested that either the second or third boundary contain the objective value. This allows you to observe how the responses are distributed (either above or below the objective value) and to decide whether to revise the objective value upward or downward.

Note

For an understanding of RTM data collection within the network control units, or their attached distributed function devices, see the relevant component description manuals.

Session Classes

Session class definitions provide a dual function. They provide the session selection criteria that determine to which session class each session belongs, and they also provide the SAW and RTM class names from which the sessions falling within each session class should take their SAW and RTM class values. Hence, unless each SAW and RTM class defined is nominated by at least one session class definition, their attributes are never used by NTS.

Session partner names are available as session selection attributes, and the definition for the class can use *wild* character positions and generic character strings as the criteria to be matched. You can therefore select a specific name, such as an application, or generic names, such as all terminals on a certain line or NCP, and any number of combinations of such names can be specified.

Other session selection criteria include:

- The Class-Of-Service name (COSNAME) for the session
- A SSCP name
- An Explicit Route (ER) number
- A Virtual Route (VR) number
- The Transmission Path (TP)
- Session type (for example, LU-LU)
- Session class (for example, XDOM)
- The source of the SAW data (SAW data may come from the local VTAM or from a remote VTAM if you are using NTS-SI)

Session classes are defined using the DEFCLASS command.

Session Classification

When NTS receives a new session start notification from VTAM, it searches your session class definitions for the best match. Session attributes are checked in the following order:

- Primary session partner name
- Secondary session partner name
- COSNAME
- ER number
- VR number
- TP

More specific attributes are checked before less specific ones; for example, the name TSO1B is checked before the generic name TSO>. Any attributes not specified are considered to be *wild*, and to match any session value.

When a match is found, NTS stores the options defined for the SAW, RTM, and resource class names (if present) with the session data. These options determine the form of subsequent processing for the session.

Next Best Match

If an RTM definition is not supplied for the class with which the session is initially matched, the search continues for an RTM class definition in the next most suitable session class.

Similarly, if a SAW definition is not supplied for the class with which the session is initially matched, the search continues for a SAW class definition in the next most suitable session class.

Session data can therefore be derived from more than one session class. Session data can, for example, include the following:

- SAW options, derived from the session class that has a matching primary session partner name
- RTM options, derived from the session class that has a matching secondary session partner name

It is also possible to have a session class definition where every attribute is *wild* (in other words, every session will match it). This enables you to supply default SAW or RTM classes for those sessions that do not match any of the more selective session classes.

Figure C-1 shows a representation of NTS class definitions, and examples of the NTS class selection process using these definitions.

Figure C-1. Session Class Definitions

Pri-name	Sec-name	COSNAME	ER	VR	TP	SAW Class	RTM Class
CICS	LCL>	*	*	*	*	CICS	CICSLCL
CICS	REM>	*	*	*	*	CICS	CICSREM
TSO	*	*	*	*	*	NOKEEP	
TSO>	*	*	*	*	*	TSO	
*	LCL>	*	*	*	*		RTMLCL
*	REM>	*	*	*	*		RTMREM
*	*	ISTVTCOS	0	*	*	NOLOG	
*	*	*	*	*	*	SAWDEF	

Session Class Definitions

In this example, names ending with > indicate that any sequence of characters can follow the prefix, and an asterisk in any column represents a *don't care* condition. Using this example, the NTS class selection process proceeds as follows:

-For a session between CICS and terminal REMA007, the SAW class name CICS and RTM class name CICSREM is selected.

-When a user logs on to TSO from a terminal named LCLB002, an initial session between TSO and LCLB002 is established. It selects the SAW class NOKEEP and the RTM class name RTMLCL. The ensuing session between the TSO target application (named, for example, TSO0019) and LCLB002 uses the SAW class name TSO and the same RTM class name, RTMLCL.

-Any session using the COSNAME ISTVTCOS and ER 0, and not involving CICS and TSO, selects the SAW class name NOLOG.

-A session between NMT and N8L4A01 on ER 1 selects the last entry, yielding a SAW class of SAWDEF and no RTM class.

NTS processes sessions according to your class definitions. See Chapter 9, *Tailoring NTS*, for information about defining classes.

RTM Class Processing

When NTS receives a session for which RTM data is to be collected, the boundary values for that class are set in the control unit, and retained for the duration of the session.

The objective response times and objective percentage for the class are used to monitor network response times, and can lead to the automatic generation of attention messages.

See Chapter 9, *Tailoring NTS*, for information about defining your RTM classes.

Resource Classification

To match a resource with a resource class definition, NTS must be aware of both the resource and its position in the hierarchy. The time when this occurs depends on the domain in which the resource is defined:

- If the resource is in the same domain as NTS, then NTS becomes aware of both the resource and its position in the hierarchy when it receives session data for a session in which the resource is participating.
- If the resource is in another domain, then NTS becomes aware of its hierarchical position only if a suitably configured Inter-System Routing (ISR) link to the NTS in the other domain exists, and the link is active.

Having been made aware of a resource and its position in the hierarchy, and providing that you have defined at least one resource class, NTS selects the resource class definition that best matches the attributes of the resource. If no class definition matches the attributes of the resource, no data is collected for it.

Resource Levels

The level of the resource class is the level of the hierarchically lowest parameter specified in the class definition. In this context, a link is at a higher level than a PU, which is at a higher level than an LU. Links are said to *own* PUs that use the link, as well as the LUs that are defined on those PUs. PUs *own* LUs that are defined on the PU.

A resource matches a resource class if all operands of parameters in the resource class definition are matched by the actual resource within the hierarchy. It is possible for a resource to match more than one resource class definition, but data from only one class definition can be stored for the resource. NTS searches resource class definitions in order from most to least specific, and selects the first resource class definition that matches the attributes of the resource.

Mechanics of Resource-matching

NTS compares resource attributes to the values of resource class definition parameters as described below:

- If an attribute of a resource matches the value of a parameter in more than one resource class definition, then NTS selects the class in which the match is most specific, in the order LU, then PU, then link.
- If higher level parameter values in a resource class definition are matched, then NTS attempts to match resources with resource class definitions at the same level.

  For example, if the resource is an LU, NTS first searches resource class definitions that have the LU parameter as the hierarchically lowest parameter, for a match.

- If the resource does not match any resource class definitions at its own level, then NTS checks to see if there is a resource class at a level above with a member that owns the current resource. If there is more than one, then NTS selects the resource class that is hierarchically closest to the current resource.

  For example, if the resource is an LU, a matching PU-level class is selected before a matching link-level class.

- A resource cannot match a resource class if the level of the resource is higher than the level of the class.

  For example, if the resource is a PU, it cannot match class definitions that have the LU parameter as the hierarchically lowest parameter.

Resource class definitions determine the way NTS processes data for different network resources or groups of resources.

For information about defining your RTM classes, see Chapter 9, *Tailoring NTS*, and Chapter 11, *Maintaining NTS*, in this manual.

Collecting Resource Statistics

Resource statistics are requested on the basis of resource classes. A resource class can specify that statistics are to be collected, and NTS will collect statistics for any resource that matches the resource class, provided that the resource statistics function is *not* disabled.

Collection Intervals

NTS collects resource statistics at specified intervals. These intervals can be thought of as discrete *buckets* into which NTS accumulates all accounting data for a particular resource during that period. When the specified interval expires, NTS resets the counters to zero and starts collecting data in a new *bucket*. After a specified (or default) number of intervals, NTS wraps, that is, overwrites the counters for the oldest interval, and so on.

Intervals provide one of the basic units for the analysis performed by the NTS Resource Statistics option.

Resource RTM Statistics

NTS attempts to collect RTM statistics for PUs (specifically, cluster controllers), if both of these conditions apply:

- Resource statistics collection is enabled (both globally and in the resource class definition).
- The resource class definition includes the name of a defined RTM class.

On the expiry of an interval, NTS solicits RTM data from a resource that meets these requirements, while simultaneously resetting the RTM counters of the resource. In this way, accurate response time data is collected for each interval.

NTS collects two sets of RTM statistics:

- *Aggregate* statistics, derived from *all* RTM responses received from a resource, irrespective of the format of the response
- *Detailed* statistics, derived from only those RTM responses received from a resource that have the exact format specified in the RTM class that matches the resource class definition

If no RTM responses received from a resource have the exact format specified in the matching RTM class, only aggregate RTM statistics are collected.

Cross-Domain Statistics

If one resource involved in a session is defined in a remote domain, then NTS can still collect statistics for the cross-domain resource, provided the following conditions apply:

- There is an ISR link between the two domains that is configured to allow unsolicited data transfer.
- Resource statistics collection is enabled in both NTS systems.

When an ISR link is established, a *handshake* occurs that allows each NTS to calculate the time difference between the system clocks. This figure is then used to calculate the completion time of the resource statistics collection interval of the other NTS. The remote NTS waits until this time before forwarding the collected statistics to the local NTS.

If no suitably configured ISR link exists, or if resource statistics collection is disabled in either NTS, no cross-domain statistics collection occurs.

Monitoring Resource Availability

If NTS has been instructed to collect statistics for a particular resource, it automatically uses SAW data to monitor the availability of that resource. A resource is considered to be *available* if it is participating in a session with the SSCP of the domain in which it is defined.

When NTS is notified by VTAM of the first session involving a resource, an SMF record is presented to the NTS User Exit, indicating that the resource is *available*. When notified of the termination of the last session in which the resource was involved (that is, the session with the SSCP), NTS presents an SMF record to the NTS User Exit indicating that the resource is *unavailable*. In addition, SMF records containing interval-based resource statistics are presented to the NTS User Exit, indicating what the current status of the resource is.

NTS Resource Statistics Logging

On the expiry of each resource statistics collection interval, NTS waits for a period of up to the correlation interval for any outstanding statistics. These statistics might consist of:

- Statistics collected for resources—owned by the local SSCP—that are participating in cross-domain sessions. These statistics will have been collected by NTS systems in one or more remote domains.
- RTM data solicited by NTS at interval expiry

NTS reports the arrival or non-arrival of statistics from other domains in the Management Services activity log. In the case of the non-arrival of statistics, the log entry specifies why statistics were not received from other domains. A separate log entry is created for each SSCP of which NTS is aware.

When this process is complete, NTS writes an entry to the activity log signaling that logging is about to commence. In this way, you can gain an accurate indication of the completeness of the statistics collected for each resource statistics collection interval.

Finally, NTS passes all the statistics collected during the interval to the NTS user exit, if one is active.

Note

Resource *statistics* are not logged to the NTS database, but passed to SMF for processing.

Collecting Further Data

When NTS is aware of active sessions and resources, further data can be collected and stored with the appropriate session records in its database.

Session and Resource Data

NTS keeps only those session records that are flagged to be kept by the matching SAW class. (The only exception is SSCP-SSCP session records, which are always kept.) Any trace, accounting, or RTM data collected by NTS is stored with the appropriate session record.

NTS stores both *resource* records and *session* records in its database. For each session record that is kept, there is always a resource record that represents the session partner. In addition, NTS keeps a record for every resource in the domain of the VTAM host system in which it is running, regardless of whether sessions with that resource are kept or not.

Note

The only way that NTS can be made aware of resources is through SAW. NTS is therefore only aware of resources that are currently active; that is, are involved in an SSCP-PU or an SSCP-LU session. NTS might have database records for additional resources, but each of those resources must have been active at some previous time when NTS was running.

Session Trace Data

NTS provides a trace monitoring capability that selects and formats trace data (which consists of copies of Path Information Units (PIUs) that flow on traced sessions) for a specific resource as it arrives from VTAM. Trace data is time-stamped by VTAM before being passed to NTS for correlation with other data related to the appropriate session.

NTS stores session establishment PIUs with session records in an initial queue. When this queue is full, or when session establishment is complete, subsequent PIUs are placed in a final queue. When the final queue is full, wrap processing occurs (that is, the oldest PIUs are deleted to make way for the newest ones). The depth of the queue is determined by the SAW class definition for the session.

Formatted trace PIUs are directed to a user's OCS screen, the Management Services activity log, or both, according to the options you specify. This facility allows an OCS operator to closely monitor a particular resource, or an NCL procedure can be written to examine the session data flow.

Resource Trace Request

When a specific resource is being traced, data is collected for all sessions that involve the resource.

If the resource is unknown to NTS when the trace request is issued, the request is nevertheless accepted and passed to VTAM. Provided that the major node to which the resource is defined is currently active, VTAM accepts the request. NTS remembers such requests, which remain in a pending state until the resource is activated.

Similarly, if a resource being traced is deactivated, NTS places the trace request in a pending state until it is reactivated, or until a trace termination request is received.

Accounting Data

NTS accounting data is extracted from the trace data supplied by VTAM and is therefore dependent on the capture of trace data. Trace PIUs, which are not relevant to accounting, are discarded unless the STRACE command is issued.

NTS can collect accounting data for selected sessions only, or globally for all sessions, depending on the value of the SYSPARMS NTSACCT operand.

Selective Collection of Accounting Data

When selective accounting (the default) is specified, accounting data is collected for those sessions that match SAW classes requiring the collection of accounting data.

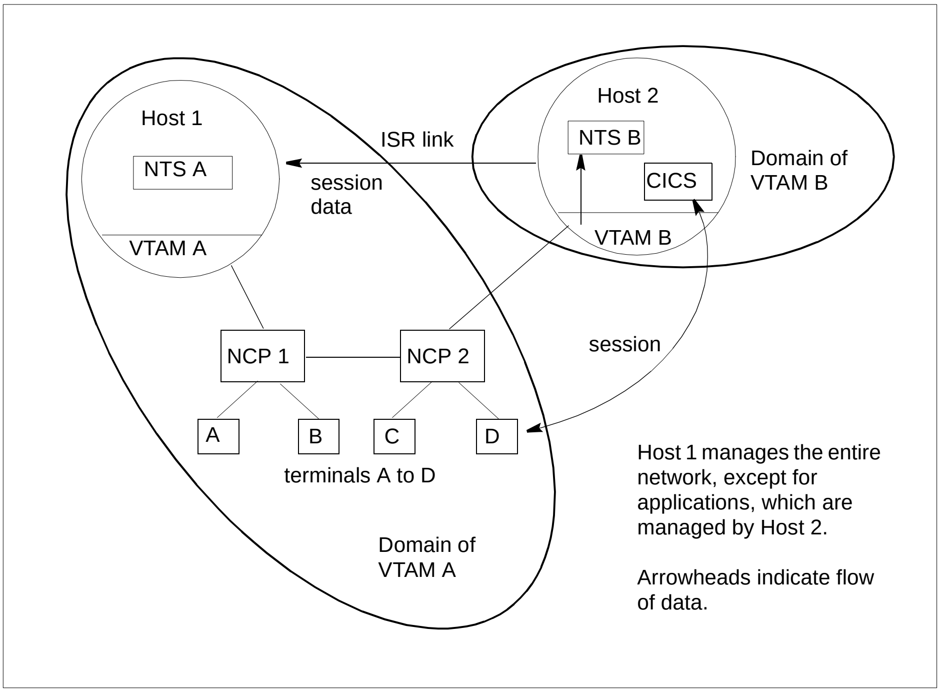
A specific trace request is issued by NTS to the session partner that resides in the VTAM subarea (or to the primary partner, if both session partners reside in the local subarea). This is because VTAM only captures trace data when the session traffic transits the VTAM host. In other words, unless one session partner resides in the VTAM subarea, no trace data is captured by VTAM.

Collection From Another VTAM Domain

The NTS Single Image (NTS-SI) feature—see *How NTS-SI Works*, on page C-14—does, however, allow you to access data from domains other than the domain in which your NTS system is active. Provided that the NTS systems are linked by suitably configured ISR links, NTS-SI allows you to access trace data from other NTS systems in other domains exactly as if the session partners were both in the local domain.

In Figure C-2, VTAM A is aware that there is an existing session between terminal D and CICS. However, because this session does not transit VTAM A, trace data is not delivered directly to NTS A. Trace data for this session is received from NTS B, due to the existence of a suitably configured ISR link.

Figure C-2. Capturing Trace Data



Host 1 manages the entire network, except for applications, which are managed by Host 2.

Arrowheads indicate flow of data.

Global Collection of Accounting Data

When global accounting is specified, NTS starts all global tracing options, in order to capture trace data for every available session. In this case, the starting and stopping of trace data collection by means of the STRACE command does not affect data capture, but only whether trace PIUs are retained or not.

Response Time Monitor Data

NTS only attempts the collection of RTM data for LU-LU sessions that are matched with an RTM class. In addition:

- The secondary resource involved in the session must be within the domain of the VTAM in the host where NTS is running.
- The PU name of the terminal control unit for the secondary device must be known to support RTM (NTS assumes that a PU supports the RTM facility, unless a response to the contrary is received).

When the session start notification is received for such sessions, the RTM class values are extracted by NTS and stored with the session record. At the same time, a request is sent to the terminal control unit to set the RTM boundary and definition parameters for the device according to these RTM class values. If the PU indicates it does not support the RTM function, or does not support host programming, then no further requests are sent to the control unit.

When a session for which RTM data is being collected ends, the RTM data collected by the secondary device is sent unsolicited to NTS and stored with other session information.

Soliciting RTM Data

RTM data can also be solicited during NTS review functions, or, more systematically, through the standard NEWS RTM procedures that allow collection on a timer or interval basis (see the *NetMaster for SNA User's Guide*). Whenever RTM data is solicited by any means, NTS updates its statistics for the session.

Analyzing RTM Data

As RTM data arrives, it is first examined by NTS to determine whether or not the performance objectives defined for the RTM class have been met. If not, this information is appended to the CNM record and made available to the NEWS feature of NetMaster for SNA. This can lead to generation of performance events and, ultimately, attention messages to notify network operators of a performance problem.

When a session for which RTM data is being collected ends, a performance event notification is also generated by NTS if the response time objectives are not met (see the section, *System Event Generation*, on page 3-8).

Data Correlation

One of the primary functions of NTS is to gather data from a number of sources and correlate it at session level. To protect NTS from waiting indefinitely for session data, you define an interval that represents the time limit for data correlation.

Enforcing the Correlation Interval

The NTS correlation interval is enforced in these situations:

- When session trace data arrives from VTAM before NTS has been notified by VTAM of the start of that session. All trace data for the pending session is kept until either the session start notification is received, or the NTS correlation interval expires (in which case it is discarded).
- While waiting for all session-related data to arrive before committing a session record for logging after the session has ended:
  - Following receipt of session end notification from VTAM, NTS determines whether any trace or RTM data is pending. If such data is expected, NTS waits for it to arrive, or for the correlation interval to expire, before continuing output processing.
  - When waiting for unsolicited data from a connected NTS system, either at session end or after a resource statistics collection interval expiry, NTS waits for it to arrive, or for the correlation interval to expire, before continuing with the logging process.

How NTS-SI Works

The NTS Single Image (NTS-SI) feature allows you to access data from domains other than the domain in which your NTS system is active. Provided that the NTS systems are linked by suitably configured ISR links, NTS-SI allows you to access trace data from other NTS systems in other domains exactly as if the session partners were both in the local domain.

It is possible to centralize the monitoring of logical network activity in multiple domains by expanding the sources of data available to an NTS system to include these types of data:

- *Session awareness (SAW) data* collected by NTS systems in other domains
- *Session data*—that is, session trace, accounting, and RTM data—collected by NTS systems in other domains

The other NTS systems may or may not be within the same SNA network. You are presented with a *single image* of the sessions between resources throughout the network, and of the performance and problem determination data collected for these sessions, provided that you do this:

- Correctly configure your NTS systems
- Correctly configure the ISR links via which your NTS systems communicate

This single image perspective is preserved in the NTS database and NTS user exit.

The user is not aware that data originates from both local and remote domains. Actions performed by NTS in response to requests to view information might differ, depending on the source of the information, but all commands, panels, and general presentation of data are consistent, irrespective of the data source.

ISR uses Inter-Management Services Connection (INMC) to convey data between systems. For further information about INMC and ISR link configuration, see the *Management Services Administrator Guide.*

NTS-SI Configuration

NTS-SI allows SAW data and session data collected in one domain to be passed on to another NTS running in another domain, provided that a direct ISR link exists between the two NTS systems.

Example

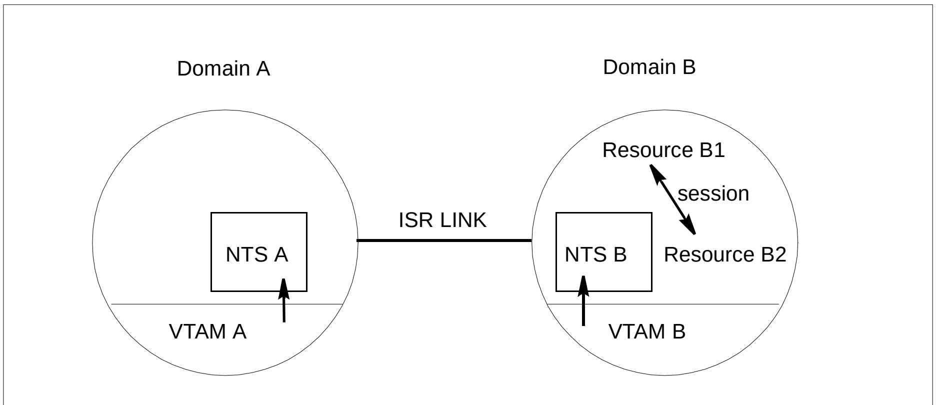
Suppose that session B1-B2 exists between two resources, B1 and B2, in domain B. In order for the NTS in domain A to be aware of SAW and session data for this session, the following conditions must be true:

- NTS must be active in both domains.
- An ISR link must exist between the domains. The link must be configured so that NTS A is specified as the receiving system (that is, the SAW operand of the ISR command is set to INBOUND) and NTS B is specified as the sending system (that is, the SAW operand of the ISR command is set to OUTBOUND).
- Class definitions in NTS B must specify that data associated with session B1-B2 is to be forwarded.
- Class definitions in NTS A must specify that data associated with session B1-B2 is to be retained.

Provided that these conditions are met, a session start notification received by NTS B from VTAM B is forwarded across the ISR link to NTS A. If any session data arrives for session B1-B2 from VTAM B, NTS B forwards an indication to NTS A that this data is available. Users of NTS A can solicit this data, as required, from NTS B. When the session ends, a session end notification received by NTS B from VTAM B is forwarded to NTS A. NTS A performs end-of-session processing for this session according to the session class definition.

A diagram of this process is shown in Figure C-3.

Figure C-3. Transfer of Session Data via an ISR Link



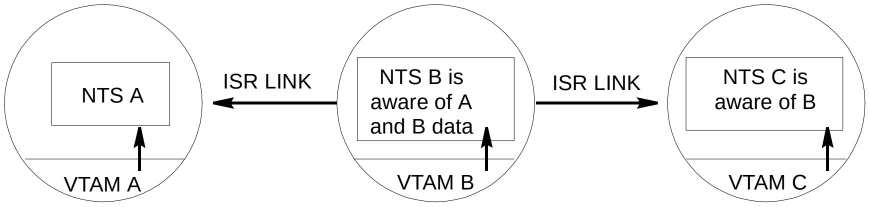
VTAM in domain B supplies NTS B with data relating to the session between resources B1 and B2, which in turn is forwarded to NTS A.

Arrowheads indicate flow of data.

Data Propagation

NTS only forwards data it has received from the local VTAM to one other NTS system. SAW and session data received via ISR from a remote VTAM are not forwarded. This is illustrated in Figure C-4. From the diagram, it can be seen that NTS B has information for sessions in domains A and B. NTS B forwards data relating to sessions in domain B to NTS C, but does not forward data relating to sessions in domain A. Therefore, when constructing a network image, NTS C cannot include SAW and session data from domain A.

Figure C-4. Data Propagation Across ISR Links



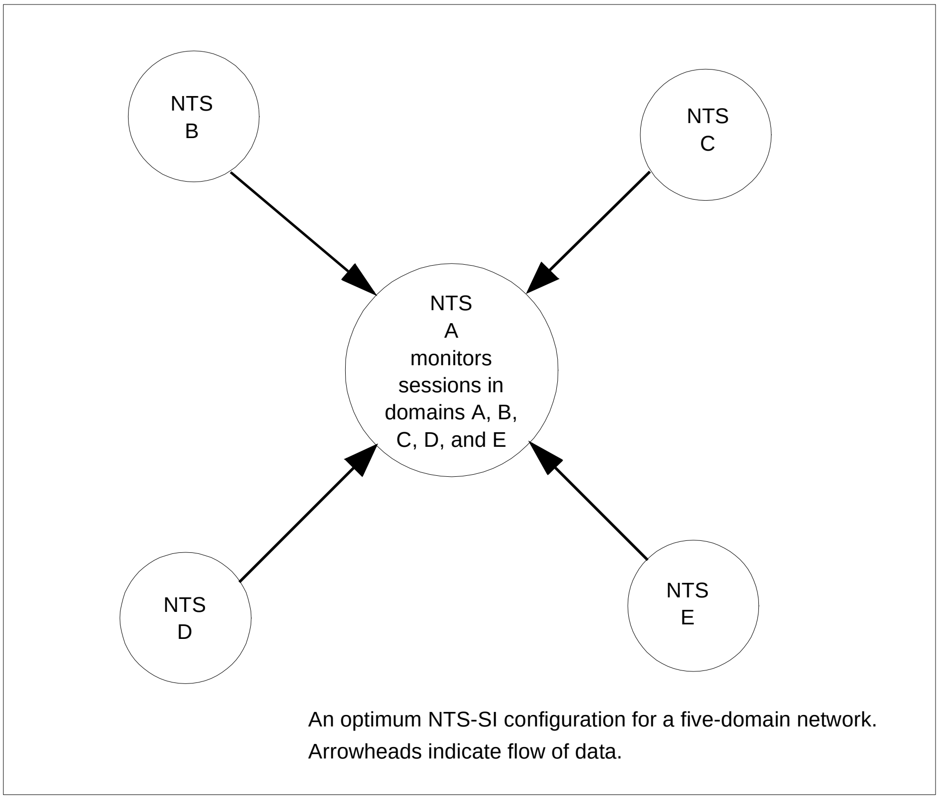
The NTS in domain C does not include data from domain A in its single image, as there is no direct A - C ISR link.

Arrowheads indicate flow of data.

Star Network Configuration

For single network image presentation, the most useful configuration of NTS systems is a star network, which enables the monitoring of network activity to be centralized (or distributed). A diagram of this configuration is shown in Figure C-5.

Figure C-5. Optimal NTS Configuration



An optimum NTS-SI configuration for a five-domain network. Arrowheads indicate flow of data.

The central (hub) NTS system monitors all network activity in both its own domain and in the outlying (spoke) domains; the spoke NTS systems monitor the activity in their own domains only. This configuration parallels the Communication Management Configuration (CMC), where a hub domain *owns* all the devices and the applications reside in the spoke domains.

How NTS Systems Share Data

Data sharing between NTS systems is controlled by the manipulation of the attributes of ISR links between the systems. The types of data able to flow across an ISR link (that is, SAW or session data, or both) and the direction of flow (inbound or outbound) are determined by the values of ISR command parameters.

Reference Network Concept

Because NTS-SI makes it possible to share SAW data between NTS systems in different networks, NTS has a *reference network* concept. Although a cross-network session is actually a single, logical connection, the session has a different appearance (due to alias names and network addresses) to VTAMs in each network. NTS commands that display or manipulate session information have a REFNET operand that allows a specific reference network ID to be specified.

Dormant NTS Concept

It is possible to start an NTS system solely for the purpose of having it receive SAW and session data via ISR links; that is, you can disable data collection from the local VTAM. This is referred to as a *dormant* NTS system.

SAW Data Sharing

NTS-SI enables NTS to obtain SAW data for sessions that are unknown to the local VTAM. There are no restrictions on which systems can share SAW data. However, for SAW data sharing between NTS systems to occur, the following conditions must be true:

- Both systems must be active.
- An ISR link must be active between the systems, with an NTS conversation currently enabled for:
  - Outbound transfer of SAW data from the sending system
  - Inbound receipt of SAW data by the receiving system

It is possible for one NTS system to be both sending and receiving SAW data at the same time, but SAW data sharing terminates if one of the conditions required for transfer is disabled.

To facilitate the operation of SAW data sharing, the ISR command supports specialized parameters that are valid for NTS conversations only. These are described in Chapter 9, *Tailoring NTS*.

SAW Data Sharing Rules

NTS systems determine which SAW data is available for sharing with other systems, on the basis of the following rules (some rules are dependent on whether the link is cross-network or cross-domain):

- **Rule 1** (applies to all link types):
  SAW data is forwarded to another NTS system only if it is not accessible to the VTAM in that domain. (NTS is able to determine whether SAW data for a particular session is visible to the VTAM in another domain.)

  The application of this rule means that no unnecessary ISR traffic is generated.

- **Rule 2** (applies to cross-domain links only):
  SAW data is available for forwarding only if it was received from the local VTAM. This means that, for an NTS system to see all network activity in a particular domain, one of the following must be true:

  - It must be in session with the VTAM in that domain.
  - It must have a direct, suitably configured ISR link with an active NTS system in that domain.

- **Rule 3** (applies to cross-network links only):
  SAW data is available for forwarding only if it was collected from the local VTAM, or from an NTS system within the same network as the local VTAM. This means that SAW data received across ISR can be forwarded to an NTS system in another network, provided that the data was derived from an NTS system *within the local network*. A corollary to this is that SAW data received from an NTS system within another network cannot be forwarded.

- **Rule 4** (applies to cross-network links only):
  A single NTS system can receive SAW data from only *one* NTS system in another network at any one time. Any attempt to enable multiple ISR links for SAW data receipt from multiple NTS systems in other networks will fail.

  The purpose of this rule is to enforce a *gateway* concept, where SAW data is sent to a central NTS system within one network before being forwarded to an NTS system in another network.

SAW Data Clean-up

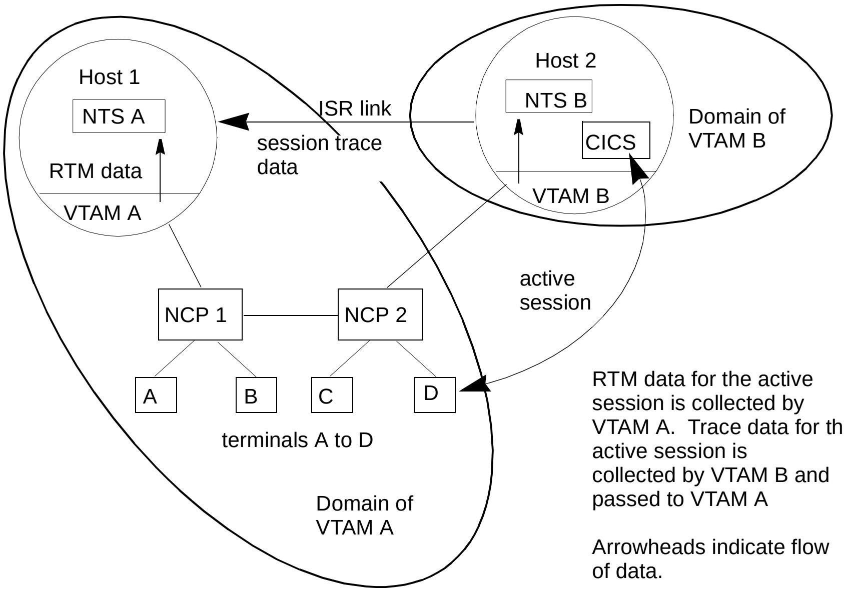
When an NTS system detects that SAW data sharing with another NTS system has terminated for any reason, it purges from storage any SAW data that was received exclusively from that system. This precaution is taken in case the data is no longer up to date. In this way, the image presented by NTS is kept accurate and current.

Session Data Sharing

Unless NTS-SI is operating, complete session data might not be available to an NTS system (even if one of the session partners is in the local domain).

For example, while cross-domain SAW data is easily accessible to NTS systems running in different domains that are linked by ISR, accounting and trace data are accessible only via the local VTAM. In addition, RTM data can be collected only in the domain in which the controller is defined. Therefore, in the case of a cross-domain session between an application and a remote terminal, one NTS system has access to the accounting and trace data, and another has access to the response time information for the same session. If you are using NTS-SI, you nevertheless have access to all the session data—accounting, trace, and response time—for any session visible to either VTAM. This is illustrated in Figure C-6.

Figure C-6. Capturing Trace Data From Another Domain



RTM data for the active session is collected by VTAM A. Trace data for the active session is collected by VTAM B and passed to VTAM A

Arrowheads indicate flow of data.

Necessary Conditions

You request session data by using the ISR command. There are no restrictions on which systems can share session data. However, for session data sharing between NTS systems to occur, the following conditions must exist:

- Both systems must be active.
- An ISR link must be active between the systems, with an NTS conversation currently enabled for these types of data transfer and receipt:
  - Outbound unsolicited transfer of data from the sending system
  - Inbound unsolicited and solicited receipt of data by the receiving system

It is possible for one NTS system to be both sending and receiving session data at the same time, but session data sharing terminates if one of the conditions required for transfer is disabled.

Session Data Sharing Rules

NTS systems determine which session data is available for sharing with other systems on the basis of the following rules (some rules are dependent on whether the link is cross-network or cross-domain):

- **Rule 1** (applies to all link types):
  Session data is forwarded to another system only if it is likely to be unavailable to the other system. In some cases, session data is visible to the two VTAMs in two different domains, and therefore to the NTS systems running in these domains.

  NTS is able to determine whether the data is visible in more than one domain, and whether the NTS in the other domain is likely to be collecting data for this session or not.

- **Rule 2** (applies to cross-domain links only):
  Session data is available for forwarding only if it was collected by the local VTAM from the local domain. Session data received from another system is *not kept in storage* but handled in one of the following ways:
  - Displayed directly as part of an NTS onscreen display
  - Logged immediately to the NTS user exit and the NTS database

- **Rule 3** (applies to cross-network links only):
  Session data received from a cross-domain ISR link can be forwarded to an NTS system in another network, *provided that* the receiving NTS system verifies that SAW data relating to this session has been forwarded to the cross-network NTS system.

This means that the *gateway* NTS system performs a routing role, to ensure that:

- Cross-network session data requests are forwarded to the NTS system that is able to respond to them (that is, the system that can collect the information directly from VTAM).
- Unsolicited session data and session data notifications are forwarded to linked networks, to provide the cross-network system with a complete picture of network activity.

Session Data Flows

Session data sharing is implemented by means of three separate transaction types or flows that can occur within the scope of a single session:

- Session data availability notifications
- Session data solicitations (both request and reply)
- Unsolicited data records at session end

These transaction types are described in the following sections.

Session Data Availability Notifications

When an NTS system receives trace or RTM data for a session for the first time, or when the NTS system becomes aware that accounting data for a session needs to be collected, it checks to see whether either of the following is true:

- SAW data sharing is in progress between the local NTS system and any other NTS systems.
- The session is cross-domain—and if it is, is there is a suitably configured ISR link to an active NTS system in the other domain.

If an NTS system accepts a data available notification, it indicates the availability of this data on any NTS Session List display.

After Session Awareness Activation

You can initiate session data sharing after session awareness has been activated in the sending system. In this case, session data might already have been collected. When the sending system detects that session data sharing with another system has become active, it sends data available notifications to this system for the sessions for which data has been collected.

On Session Termination

If an NTS system detects that session data sharing with another system has been terminated for any reason, it resets the session data present indicators that were set exclusively in response to session data available notifications received from that other NTS system. This is because the data can no longer be solicited.

Session Data Solicitation

After an NTS system has received notification of data availability, it can send a solicitation request to the collecting system to view all or part of the data. This occurs when a user requests a particular display. The solicitation requests that the collecting system immediately forward a reply containing all collected data of the specified type.

Session data received in reply to a solicitation request is displayed immediately. When the user exits the display, the data is discarded. Another user request to view the session data results in another solicitation. Refreshing the current display also causes the current data to be discarded and another solicitation to be issued. In this way, NTS guarantees that the data displayed is the most recent (and therefore most accurate) available, and that the *data is actually stored in only one location in the network*.

Unsolicited Session Data at Session End

When an NTS system detects the end of a session, the following processing occurs:

- The NTS system determines whether there are any other NTS systems that are aware of the session but do not have actual visibility of the session data. If this is the case, then the NTS system forwards locally sourced data that is not visible (or not being collected) in the other domain, to the other NTS system. In this way, the complete data for the session is made available to the other NTS user exit and database.
- The NTS system determines whether there is any session data type that has been requested but has not yet arrived. If this is the case, then the NTS system determines whether it has any suitably configured ISR links to any systems that could provide the data and waits for a limited period of time for the data to arrive via the ISR link or links.

When session data arrives from another NTS system, the receiving system determines whether this session data was requested or not. Any data that was not requested is discarded. If the data was requested, then it is immediately logged to the NTS database or user exit (or both, depending on what is requested) and the session and accompanying data is purged from storage.

D

NEWS Device Solicitation Procedures

You can solicit data from network devices using the NEWS Device Support option from an OCS window, or on an unattended basis using the AT and EVERY commands. The NEWS command procedures described in this chapter are a means of passing solicited data to other procedures in an automated operations environment, so that these procedures can act upon the data returned.

When a request for data has been successfully completed and the response returned to the user, the format of the response is similar to that used by VTAM (that is, each line of data is returned as a message with a message number that is unique to that data).

This chapter provides a brief guide to the NEWS procedures available for soliciting data from an OCS window, or on an unattended basis. For each procedure, a description is given of the parameters required for operation and the type of data solicited.

For further details about the procedures, see the comments at the start of each procedure.

Line Command Procedures

The NEWS line command procedures are intended primarily as a means of passing solicited data to other procedures in an automated operations environment so that they can act upon the data returned.

The procedures, when called, have a similar format to those of current commands and the parameters are, overall, keyword driven.

$NW386SO
Solicits link status or DTE test results from a 386x type modem configured with the LPDA-1 option.

$NWDS13B
Provides a batch command interface for the Central Site Control Facility (CSCF).

$NWFCSSO
Solicits loop status, loop errors, and response time data from an IBM 3600/4700 Financial Communication System devices. The data is always returned to CNMPROC for logging.

$NWLPDA2
Records or changes, configuration or coupler parameters; changes the modem’s functional characteristics; or runs online diagnostic tests for an LPDA-2 device.

$NWRTMSO
Solicits RTM data from a 3x74 controller that supports the RTM function. RTM data may be requested for a single LU, or for only those LUs with non-zero data.

$NWRUNCM
Packs a command into an NMVT RU to be sent to, and executed by, a service-point application. Responses received are displayed as text messages.

$NWSOLCT
Solicits secondary end errors and engineering change level data from a PU that supports such requests. Secondary end errors include link test statistics, summary error data, communications adapter error statistics and EC level information. Any or all of these summaries may be requested.

$NWVPDSO
Solicits vital product data from a PU (and its port-attached devices).

Each device solicitation procedure is described in full in the following sections.

$NW386SO

Function

This procedure solicits link status or DTE test results from a 386x type modem configured with the LPDA-1 option. You can also obtain this data by selecting options 6 or 7 from option G of the NEWS Device Support menu.

```
$NW386SO      NODE=network_name
              REPORT={ LINK | DTE }
             [ NCP=NCP_name ]
             [ RESET={ YES | NO } ]
             [ LINK=link_name  |
              SSCP=SSCP_name ]
```

Use

Use this procedure to solicit link status or DTE test results for a type 386*x* LPDA-1 modem from a nominated node/NCP. The data is always returned to CNMPROC for logging.

Operands

NODE

The network name of the device from which the data is solicited.

REPORT= LINK | DTE

Specifies the type of data required:

LINK

Link status test.

DTE

DTE test.

NCP

The name of the NCP owning the specified node.

RESET=YES | NO

Specifies whether the counters in the controller are to be reset after solicitation has completed.

LINK

The name of the Management Services link to the system in which the node name is located.

SSCP

The name of the SSCP controlling the node.

Example

```
$NW386SO NODE=TSTM386 NCP=NCP01 REPORT=DTE
```

This example results in a request for a DTE test to be performed at node TSTM386 controlled by NCP NCP01.

$NWDS13B

CSCF Batch Command Interface

The CSCF Batch Command Interface enables the execution of all CSCF functions in a batch NCL mode. This enables you to control a controller through automation, so that an end user does not have to be logged on to execute controller functions. The command interface is the execution of the $NWDS13B procedure with parameters dictating processing flow.

$NWDS13B FUNC = LOGON

```
&CONTROL SHRVARS=($NWCS#,$NW#USR,$GP) NOVARSEG

$NWDS13B        FUNC=LOGON
               { NODE=node_name }
               [ OP=command line text ]
               [ LINK=link_name ]
               [ SSCP=sscp_name ]
               [ PRINT={ YES|nnn|LOG } ]
```

Note

Print is skipped if RETCODE greater than 4 occurs.

$NWDS13B FUNC = ACTION

```
&CONTROL SHRVARS=($NWCS#,$NW#USR,$GP) NOVARSEG

$NWDS13B       FUNC=ACTION
              { KEY=PFnn|ENTER }
              [ OP=command line text ]
              [ DATA1-n=xx ]
              [ PRINT={ YES|nnn|LOG } ]
```

Note

Print is skipped if RETCODE greater than 4 occurs.

$NWDS13B FUNC = LOGOFF

```
&CONTROL SHRVARS=($NWCS#,$NW#USR,$GP) NOVARSEG

$NWDS13B     FUNC=LOGOFF
```

Parameter Descriptions

FUNC

Indicates to the batch procedure what process to take. This is a required parameter where the valid values must be:

$NWDS13B FUNC = LOGON

LOGON the session ID; must be first call.

$NWDS13B FUNC = ACTION

Indicates that some type of action is to take place.

$NWDS13B FUNC = LOGOFF

LOGOFF the session ID after processing is completed.

NODE

The 3174 node name, required only on the logon call.

OP

The command to be entered. This represents what would be the command line on an on-line panel; that is, any valid CSCF command or option (for example /5,2). The F key command text is not supported at the command line (RETURN, for instance).

KEY

The key (one of PF1 to PF24, or ENTER) that is to be entered once at an indicated panel.

DATA1-20

The data to be entered for each input field on a panel. If the third data item on a panel is to be updated, then DATA3 is passed with the data to be entered.

LINK

The link name if remote controller operations are desired only on the logon call.

SSCP

The SSCP name if remote controller operations are desired only on a logon call.

PRINT

Specify YES to print screen using user's default PSM printer ID; or specify a specific printer; or specify LOG to send screen capture to the Management Services log. This option is skipped if a non-zero RETCODE occurs. The PRINT will occur after all OP and/or KEY parameters are processed.

Return Variables

Data returned to calling procedure for interrogation is as follows:

&RETCODE

0

Batch process completed. This simply means that a command was sent to the controller and a response was received from the controller.

4

Key supplied is not active on current panel.

8

Processing error.

12

Unable to log on.

16

Invalid parameters.

&SYSMSG

ERROR MESSAGE.

&$NW#USR#L*nn*

Variables where L*nn* represents up to 24 lines of panel data.

Examples of Batch Command Usage

Example 1: IML the controller.

```
EXEC $NWDS13B FUNC=LOGON NODE=ACSC11
EXEC $NWDS13B FUNC=ACTION OP=14 KEY=ENTER
EXEC $NWDS13B FUNC=ACTION OP=1,2,41 KEY=ENTER
EXEC $NWDS13B FUNC=ACTION OP=password KEY=ENTER
```

Example 2: Reset event logs, cable errors and trace data.

```
EXEC $NWDS13B FUNC=LOGON NODE=ACSC11 OP=/4,2
EXEC $NWDS13B FUNC=LOGOFF
```

Example 3: Change configuration data (update controller vital product data).

```
EXEC $NWDS13B FUNC=LOGON NODE=ACSC11 OP=/5,2
EXEC $NWDS13B FUNC=ACTION KEY=PF04 DATA3=CINCY
EXEC $NWDS13B FUNC=LOGOFF
```

Example 4: Display configuration data and print out to specified printer.

```
EXEC $NWDS13B FUNC=LOGON NODE=ACSC11 OP=/2,2 PRINT=U33
EXEC $NWDS13B FUNC=ACTION KEY=PF8 PRINT=U33
EXEC $NWDS13B FUNC=ACTION KEY=PF8 PRINT=U33
EXEC $NWDS13B FUNC=ACTION KEY=PF8 PRINT=U33
EXEC $NWDS13B FUNC=ACTION KEY=PF8 PRINT=U33
EXEC $NWDS13B FUNC=LOGOFF
```

Note

Two methods are available when selecting options from the command line. You can either specify just the option number and get to the desired option one screen at a time, or you can take a fast path by using the forward slash (/) as a means of panel skipping. Example 1 above shows the one-screen-at-a-time route; all other examples show the fast route. See sample procedure $SANWCSF as a working example.

It is a requirement that data with embedded blanks be assigned to the DATA*n* keyword via a variable. For example:

```
&LOCATION = &STR Cincinnati, Ohio

DATA5=&LOCATION
```

As noted above, &CONTROL NOVARSEG must be in effect or a RETCODE 16 (SYSMSG `EWKB01 invalid keyword parameter`) occurs.

Note

In example 1, there is no need for a LOGOFF call as the session is implicitly terminated by the IML process.

$NWFCSSO

Function

This procedure solicits loop status, loop errors, response time data, or all of these from an IBM 3600/4700 Financial Communication System device. You can also obtain this data by selecting any of the first four options from option 3 of the NEWS Device Support menu. The data returned is always delivered to CNMPROC for logging.

```
$NWFCSSO      NODE=network_name
             [ REPORT={ STATUS | ERRORS | RESPTIME | ALL } ]
             [ RESET={ YES | NO } ]
             [ LINK=link_name |
               SSCP=SSCP_name ]
```

Operands

NODE

The network name of the device that the request is to be sent.

REPORT={ STATUS | ERRORS | RESPTIME | ALL }

Specifies the type of data required.

ERRORS = Loop errors

RESPTIME = Response time data

STATUS = Loop status

ALL = All of the above

RESET = { YES | NO }

Indicates whether or not the counters in the controller are to be reset after solicitation.

LINK

Is the name of the Management Services link to the system in which the node name is located.

SSCP

Is the name of the SSCP controlling the node to be specified.

Example

```
$NWFCSSO NODE=FCS00001 REPORT=ERRORS LINK=TEST01
```

This example results in a request for loop errors to be sent to controller FCS00001 across the link TEST01.

$NWLPDA2

Function

This procedure records or changes configuration or coupler parameters, changes the functional characteristics of the modem, or runs on-line diagnostic tests for an LPDA-2 device. You can also obtain this data by selecting option 7 of the NEWS Device Support menu. The following operands are required to execute this procedure.

```
$NWLPDA2    { DISPLAY={ CONFIG | COUPLER } |
              CHANGE={ CONFIG | COUPLER } |
              SPEED={ FULL | BACKUP } |
              DIAL [ =(num1,num2,prefix) ] |
              DISC | CONTACT={ OPEN | CLOSE | QUERY } |
              TEST={ LA | MS | TR(n) } }
              STATION=node_name
              NCP=ncp_name
            [ LEVEL={ 1 | 2 } ]
            [ MODEM={ LOCAL | REMOTE | BROADCAST } ]
            [ FILE={ YES | NO | ONLY } ]
            [ LINK=link_name | SSCP=SSCP_name ]
```

Operands

DISPLAY = { CONFIG | COUPLER }

Displays the modem's configuration parameters or the coupler's configuration parameters for the modem

CHANGE = { CONFIG | COUPLER }

Changes the modem's configuration parameters or the coupler's configuration parameters for the modem.

> **Note**
>
> The CHANGE operand is valid only if executed from a full-screen environment and FILE=ONLY is not specified

SPEED = { FULL | BACKUP }

Sets the modem's transmission speed to FULL or BACKUP.

DIAL [= (*num1*, *num2*, *prefix*)]

Establishes SNBU connections using the phone number(s) stored in the configuration field(s) or the supplied prefix and extension(s). If the prefix and extension(s) are supplied, then the total length of the numbers (including pauses) must not be greater than 41 characters.

DISC

Disconnects the line at the remote modem.

CONTACT = { OPEN | CLOSE | QUERY }

Opens or closes the modem's built-in relay or reports the status of the built-in relay (open or closed) and whether or not electric current is flowing through the sensor.

TEST = { LA | MS | TR(*n*) }

Performs modem diagnostic tests as described by the following options:

LA

Performs a line analysis test.

MS

Performs Modem and Line Status test or Modem Self-test.

TR(*n*)

Performs a test-pattern exchange between the local and remote modem to determine the line quality and number of transmission errors. *n* is the number of sequences of 16 blocks of data to be exchanged. *n* + 1 sequences are sent.

STATION=*nodename*

The network name of the device that the request is to be sent to.

NCP=*ncpname*

The name of the NCP in which the station is located.

LEVEL = { 1 | 2 }

Determines where the command is to be sent. The primary link is indicated by 1 and the tailed link by 2.

MODEM = { LOCAL | REMOTE | BROADCAST }

Identifies the type of modem to receive the command:

LOCAL

Indicates the command is to be sent to the local modem.

REMOTE

Indicates the command is to sent to the remote modem.

BROADCAST

Indicates the command is to be sent to all secondary modems. This parameter is valid only when used in conjunction with the SPEED and DISC operands.

FILE = { YES | NO | ONLY }

Directs how the returned results of the command are to be processed.

YES

Indicates the results of the command are to be displayed and sent to CNMPROC.

NO

Indicates the results of the command are to be displayed and not sent to CNMPROC.

ONLY

Indicates the results of the command are to be sent to CNMPROC only.

LINK

The name of the Management Services link to the system in which the node name is located.

SSCP

The name of the SSCP controlling the node to be solicited.

Examples

```
EXEC $NWLPDA2 DISPLAY=COUPLER STATION=STATION1 NCP=NCP1+
                                               FILE=YES
EXEC $NWLPDA2 SPEED=FULL STATION=STATION1 NCP=NCP1+
                                               MODEM=REMOTE
EXEC $NWLPDA2 CHANGE=CONFIG STATION=STATION1 NCP=NCP1
```

Return Variables

Data returned to the calling procedure for interrogation is as follows:

&RETCODE

0
Batch process completed. This simply means that a command was sent to the controller and a response was received from the controller.

8
Processing error.

$NWRTMSO

Function

This procedure solicits RTM data from a 3x74 controller that supports the RTM function. You can also obtain this data by selecting option 2.2 from the NEWS Device Support menu. You can request RTM data for a single LU, or for only those LUs with non-zero data. The following operands are required to execute this procedure.

```
$NWRTMSO      NODE=network_name
             [ LU={ ALL | 2 ...  255 } ]
             [ RESET={ YES | NO } ]
             [ RESPONSE={LOG | USER | BOTH } ]
             [ LINK=link_name | SSCP=SSCP_name ]
```

Operands

NODE
The network name of the device that the request is to be sent.

LU={ ALL | 2 ..... 255 }
Indicates if all LUs are to be solicited or only the LU specified by number.

Note
If an LU number is specified it must be in the range 2 to 255.

RESET={ YES | NO }
Indicates whether or not the counters in the controller are to be reset after solicitation.

RESPONSE={ LOG | USER | BOTH }

Indicates where the responses are to be delivered:

LOG

Indicates response data is to be sent to CNMPROC.

USER

Indicates response data is to be returned to the requesting procedure.

BOTH

Indicates response data is to be delivered to both CNMPROC and the requesting procedure.

Note

If NTS is active in the system sending the request, then the LU name can be substituted for the node name on the NODE= operand, and the LU= operand ignored (that is, if data from only one LU is required).

LINK

The name of the Management Services link to the system in which the node name is located.

SSCP

The name of the SSCP controlling the node to be solicited.

Example

```
$NWRTMSO NODE=RTMNODE1 LU=ALL RESET=NO RESPONSE=BOTH
```

This example results in a request for RTM data for ALL LUs to be sent to controller RTMNODE and the responses to be delivered to both CNMPROC and the soliciting procedure.

$NWRUNCM

Function

This procedure sends a command to be executed by a service-point application. Responses received are displayed as text messages.

```
$NWRUNCM      NODE=service_point_name
             [ LINKNAME=link_name | SSCP=remote_sscp ]
              APPL=application_name
              DATA=command_text
```

Operands

NODE

The name of the service point (PU) for executing the command.

LINKNAME

Optionally specifies the ISR link name for routing the request to a remote host which is the focal point for the NODE specified and will act as the source of the application command. If both LINKNAME and SSCP are omitted, then the request is sent from the local host.

SSCP

Optionally specifies the name of a remote host which is the focal point for the NODE specified and will act as the source of the application command. If both LINKNAME and SSCP are omitted, then the request is sent from the local host.

APPL

The name of an application residing on the specified NODE which is to execute the command.

DATA

The command text intended for the application. It must be specified as the last operand to the $NWRUNCM procedure. The SNA-imposed limit on this is 253 characters.

Examples

Example 1: From Command Entry

```
$NWRUNCM NODE=ASYD61 APPL=NETWARE DATA= +
     SNAME=RESEARCH Query Volume USpaceAllowed +
     VolName=SYS UserName=user01
```

Responses are written to the Command Entry screen:

```
EW0019 NODE=ASYD61, DATE=....
EW0020 NTWK=SDINET1, SSCP=....
EWR003 MESSAGE TEXT
EWR004 SNAME=RESEARCH USERNAME=USER01 VOLNAME=SYS
EWR004 USPACEALLOWED=10000KBYES
EW0018 *END*
```

Example 2: From an NCL procedure

```
&INTCLEAR
&INTCMD $NWRUNCM NODE=ASYD61 APPL=NETWARE DATA= +
          SNAME=RESEARCH OP=user01 Remove File Trustee+
          Path=SYS:USER01\TEST UserName=user02
&MSGNO =
&DOWHILE .&MSGNO NE .EW0018
     &INTREAD STRING=(RSPMSG)
     &PARSE VARS=MSGNO REMSTR=MSGTXT +
          DATA=&RSPMSG
     ...
     ...
&DOEND
```

$NWSOLCT

Function

This procedure is used to solicit secondary end errors and engineering change level data from a PU that supports such requests. Secondary end errors include: link test statistics, summary error data, communications adapter error statistics, and EC level data. Any or all of these summaries can be requested. You can also obtain this data by selecting any of the first five options for option 1 of the NEWS Device Support menu.

```
$NWSOLCT     NODE=network_name
            [ REPORT={ LINK | SUMMARY | COMMS | EC | ALL }]
            [ RESET={ YES | NO } ]
            [ RESPONSE={ LOG | USER | BOTH } ]
            [ LINK=link_name |
              SSCP=SSCP_name ]
```

Operands

NODE=*aaaaaaaa*

The network name of the device that the request is to be sent.

REPORT={ LINK | SUMMARY | COMMS | EC | ALL }

Specifies the type of data required:

LINK

Link test statistics.

SUMMARY

Summary error data.

COMMS

Communications adapter error statistics.

EC

Engineering change level data.

ALL

All of the above.

RESET={ YES | NO }

Indicates whether or not the counters in the controller are to be reset after solicitation.

RESPONSE={ LOG | USER | BOTH }

Indicates where the responses are to be delivered.

LOG

Indicates response data is to be sent to CNMPROC.

USER

Indicates response data is to be returned to the requesting procedure.

BOTH

Indicates response data is to be delivered to both CNMPROC and the requesting procedure.

LINK

The name of the Management Services link to the system in which the node name is located.

SSCP

The name of the SSCP controlling the node.

Example

```
$NWSOLCT NODE=TSTC01
```

This example results in a request for link test statistics, summary error data, communication adapter error statistics, and EC level data to be sent to controller TSTC01, and in the results being processed by CNMPROC.

Notes

If REPORT=EC is specified then you can only use LOG for the RESPONSE= operand.

When REPORT=ALL is specified, Engineering Change level data is always sent only to CNMPROC, no matter what option has been chosen for the RESPONSE= operand.

If REPORT=EC is specified, and the PU to be solicited is a 3174 or equivalent, then you must issue the command twice to solicit all EC and RPQ data.

$NWVPDSO

Function

This procedure solicits vital product data from a PU (and its port-attached devices).

```
$NWVPDSO      NODE=network_name
             [ REPORT={ PU | ALL } ]
             [ RESPONSE={ LOG | USER | BOTH | FILE }
             [YEARFMT={ YY | YYYY }
             [ FILEDD=ddname ]
             [ LINK=link_name |
               SSCP=SSCP_name ]
```

Operands

NODE

The network name of the device that the request is to be sent.

REPORT={ PU | ALL }

Specifies whether or not the data is to be sent for only the PU or both the PU and its port-attached devices (if the PU supports such a request).

PU

The product data is only from the PU.

ALL

The product data is from both the PU and all port-attached devices.

RESPONSE={ LOG | USER | BOTH | FILE }

Indicates where the responses are to be delivered:

LOG

Indicates response data is to be sent to CNMPROC.

USER

Indicates response data is to be returned to the requesting procedure.

BOTH

Indicates response data is to be delivered to both CNMPROC and the requesting procedure.

FILE

Indicates response data is to be written to the file specified by the FILEDD=*ddname* operand.

YEARFMT={YY | YYYY}

(Relevant only if RESPONSE=LOG is specified.) Indicates the date format of record keys for records written to the output file. Data fields are not affected.

YY

Indicates the date is to be in the format YY/MM/DD.

YYYY

Indicates the date is to be in the format YYYYMMDD.

FILEDD=*ddname*

Specifies the DD name of the file to which the RESPONSE data is to be written. It is assumed that this file has already been allocated to Management Services but not opened, and is in standard UDB format.

> **Note**
>
> The FILEDD operand is valid only when RESPONSE=FILE is also used.

LINK

The name of the Management Services link to the system in which the node name is located.

SSCP

The name of the SSCP controlling the node to be solicited.

Examples

Example 1

```
$NWVPDSO NODE=PU374501 REPORT=PU LINK=TEST01
```

This example results in a request for vital product data for only the PU to be sent to the PU PU374501 across the link TEST01 and the results to be returned to CNMPROC.

Example 2

```
$NWVPDSO NODE=TSTC02 RESPONSE=FILE FILEDD=VPDFILE
```

This example results in a request for vital product data to be sent to device TSTC02 and all its port-attached devices with the result of the solicitation (if successful) being written to the file allocated by the DD name VPDFILE.

File Format for the Vital Product Data File

Key

NODENAME (8 chars)

UNIQUE PORT NUMBER (3 digits)

YY/MM/DD (8 chars) (by default)
or
YYYYMMDD (if YEARFMT=YYYY has been specified)

HH:MM:SS (8 chars)

Fields

> **Note**
> It is probable that not all fields will be used for all records contained in this file.

1. Device hierarchy in the standard NEWSFILE record format
2. Hardware Common Name
3. Hardware Machine Type (and model (MODEL *xxx*))
4. Hardware Serial Number
5. Hardware Repair ID
6. Emulated Hardware Machine Type (and model (MODEL *xxx*))
7. Microcode EC Level
8. Software Product Common Name
9. Software Product Common Level (V*x.x.x*)
10. Software Product Program Number
11. Software Serviceable Component
12. Software Serviceable Component Release Level (*xxx*)
13. Software Customization
14. Software Customization Date and Time (YY/MM/DDHH:MM)
15. Primary LU Address
16. Hardware Group
17. Port Type
18. Port Number
19. Vendor ID
20. Physical Location
21. LAN Universal Address
22. Additional Attribute Label
23. Additional Attribute Data

E

Implementing the NEWS User Exit

NEWS can present all records received across the VTAM CNM interface to an installation-supplied user exit before any processing is performed for the record.

The exit can perform any desired processing of the record, and can indicate that the record is to be ignored by NEWS, unless it was generated in response to a solicitation request by an &CNMSEND statement.

Note

APPN alerts sent to the ALERT-NETOP NEWS application, and records from remote systems arriving over ISR links are *not* passed to the exit.

This appendix provides the following information:

- Details of the parameters passed to the NEWS user exit and the return codes required from the exit
- Details of NEWS SMF record formats

Sample NEWS Exits

Two sample NEWS exits are supplied as working models that can be modified as required. The exits are members of the *?dsnq*.SN400.SNSAMP library distributed on the installation tape.

- NEWSEXIT takes a copy of each CNM record received and writes it to a sequential dataset.

  Any DD cards required by the exit to write the CNM records to a dataset should be included in the Management Services execution JCL. The NEWSEXIT sample exit provided writes all records to a variable blocked dataset which requires a DD card with a DD name of DDNEWS.

- NEWSXSMF takes a copy of each CNM record and formats an SMF record which can then be processed by external packages that use SMF data.

  The Assembler macro $NMSMF, distributed in the Management Services macro library, provides mapping for the format of the SMF record generated. For details of SMF record formats, see the section, *NEWS SMF Record Formats*, on page E-8.

How the NEWS Exit Is Called

Note

> To implement a NEWS exit, you must define the exit name in the CNM parameter group in ICS. See the *Unicenter NetMaster Network Management for SNA Implementation Guide*.

NEWS is initialized when Management Services is started and a subtask that acts as the driver for the exit is attached. The subtask mainline routines handle communications between the subtask and the NEWS components in the Management Services mother task.

When the subtask is attached, subtask mainline routines:

1. Load the load module specified as the installation-supplied user exit.
2. Call the exit via conventional branching and linking.
3. Pass an initialization parameter list to the exit.

Processing the NEWS Exit

Because the user exit executes as part of a Management Services subtask, no restrictions are placed on the functions that the exit can perform. This is because the activities of the subtask do not impact the performance of the Management Services main task, and the exit subtask runs at a lower dispatching priority than the main task.

Processing of CNM records by the Management Services mother task is, however, delayed by processing occurring within the CNM exit.

NEWS Exit Coding Requirements

This section describes the coding requirements for NEWS exits.

Maintaining Registers on Entry to an Exit

You must observe standard linkage conventions when coding an exit.

On entry, the registers must be saved in the caller’s save area, and on exit, restored (except R15, which must contain a return code).

On entry, register contents are as follow:

R0	Unpredictable
R1	Address of a parameter list
R2-R12	Unpredictable
R13	Address of the caller’s standard save area
R14	Caller’s return address
R15	Address of the entry point of the user exit

On exit, the same registers should be restored except for R15, the exit return code. For details of the format of a return code, see the section, *About Exit Function Codes*, in this appendix.

Parameter List Format

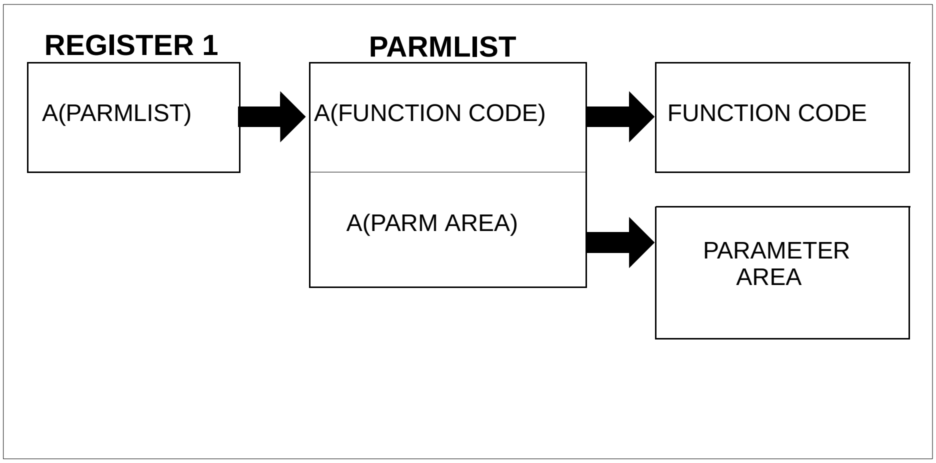
The parameter list addressed by Register 1 on entry consists of one or two contiguous fullwords. If there are two fullwords, the first contains the address of another fullword that holds a function code indicating the reason that the exit is being called.

The second fullword of the parameter list is present only for function codes 0 and 4. The high-order bit of this last (or solitary) fullword is not set, and the length of the parameter list must therefore be determined by examination of the function code.

- For function code 0, the second word of the parameter list contains the address of an area that contains system data that might be of value to the exit in determining its processing options.
- For function code 4, the second word of the parameter list contains the address of the CNM record being passed to the exit for examination.
- For function code 8, the parameter list contains one word only.

The structure of the parameter list pointers is shown in Figure E-1.

Figure E-1. Structure of NEWS User Exit Parameter List



About Exit Function Codes

The function code contained in a fullword addressed by the first word of the parameter list is a binary value that is right-aligned, with all high-order bits set to zero. Function codes used are:

- X‘00000000’ = initialization call
- X‘00000004’ = CNM record available
- X‘00000008’ = termination call

The processing associated with these function codes is detailed in this section.

Function Code 0

This indicates that the system has just been initialized. Any initialization processing that the exit needs to do, such as opening required datasets, should be done now.

The second word of the parameter list passed to the exit for function code 0 contains the address of an area, formatted as described below.

Bytes

00:03

In this full word, the exit can store the address of a message that is to be logged to the Management Services activity log and sent to Monitor class operators. On return from any call to the exit, Management Services checks this word; if the value is non-zero, it is assumed to be the address of a message. (See the section, *Issuing Messages from the NEWS Exit*, on page E-7 for the format of the messages.)

04:04

Operating system indicator. Values are:

X‘02’

MSP, MSP/AE, MSP/EX

X’10’

OS/390 or z/OS

05:05

Management Services SMF record identifier set by the SYSPARMS SMFID= command. This field is set to X‘00’ in VM/SP systems.

06:17

Management Services system identifier set by the SYSPARMS ID= command.

On successful completion of processing, the exit returns control to the caller with a return code of 0 in register 15. Any other value in R15 is regarded as indicating that processing was unsuccessful and the exit subtask is terminated abnormally and assigned User Abend reason code 390-01.

Function Code 4

This indicates that a CNM record has been received by NEWS across the VTAM CNM interface. For this function code only, the second word of the parameter list contains the address of the CNM record received. The actual CNM record is prefixed by a length field, so the format of the record presented to the exit is described in this section.

Bytes

00:01
Length of record including this 4-byte prefix

02 :03
Always X‘0000’

04 :*nn*
Length of CNM record data (variable)

The processing performed by the exit on the record received is unrestricted, but it should be noted that extensive delays in processing might cause NEWS to reach internal queue limits and result in CNM records being lost.

Note

The record presented to the exit is only a copy of the record received by NEWS. No modification can be made to the record actually processed by NEWS when the exit returns control.

The record with its length field prefix is suitable for writing to a variable blocked dataset.

When control is returned by the exit, R15 must be set to one of the following (decimal) return codes:

0 Processing complete. NEWS continues processing this record and passes the next CNM record to the exit when it arrives.

4 Processing complete. NEWS ignores this record and passes the next CNM record to the exit when it arrives. Records that have this return code are not passed to CNMPROC.

8 Processing complete. NEWS continues processing this record but makes no further calls to the exit.

12 Processing complete. NEWS ignores this record (unless it is a solicited response) but makes no further calls to the exit.

Any other value in R15 is regarded as indicating unsuccessful processing and the exit subtask is abnormally terminated and assigned User Abend reason code 390-02.

Translate-Inquiry RUs

Although TR-INQ RUs (Translate-Inquiry RUs) are passed to the user exit, the return code set by the exit for these RUs is not checked. Processing of TR-INQ RUs proceeds by the Alias Name Translation Facility of NEWS being called to format a TR-REPLY RU.

Function Code 8

This indicates that Management Services is terminating. It alerts the exit to perform any cleanup processing required, such as closing datasets. The termination of Management Services cannot proceed until the exit returns control.

On successful completion of processing, the exit returns control to the caller with a return code of 0 in register 15. Any other value in R15 will be regarded as indicating unsuccessful termination and the exit subtask is abnormally terminated with User Abend reason code 390-03.

Issuing Messages from the NEWS Exit

You might require the exit to communicate with Management Services operators to notify them of particular conditions that have been detected.

You can use the exit to generate message text and place its address in a fullword contained in the area addressed by the second fullword of the initialization call parameter list. (For a description of the initialization call, see the section, *How the NEWS Exit Is Called*, on page E-2.)

A message can be generated following any call to the exit, and its address placed in this fullword. The message must be formatted as follows:

Bytes

00:01	Length of message text (excluding these 2 bytes)
02:*nn*	Message text.

The maximum message length is 130 bytes. Excess length is ignored and the message truncated.

NEWS SMF Record Formats

The NEWS SMF exit program (NEWSXSMF) is provided as a sample user exit that receives a copy of every NEWS CNM record and writes each record to the SMF log file.

NEWSXSMF writes data to the SMF log file in two different formats:

- CNM records have a header followed by the CNM record section.
- 4700 Support Facility (TARA) data have a header followed by one or more sections containing statistical information. Such data is sent from 36xx/47xx Finance Communications Systems (FCS) devices and contained in RECFMS type X‘04’ records.

You must tailor NEWSXSMF to indicate which format is to be used, depending on the type of data you want to collect. You can indicate one of these options:

- Write all CNM records out in the CNM record format.
- Write only TARA data in TARA data format, and ignore all other records.
- Write TARA data in TARA data format, and write all other records in the CNM record format.

For information on how to tailor the NEWSXSMF program, see the comments provided within the program.

The macro $NMSMF is distributed with Management Services. It defines a DSECT describing the contents of the SMF records.

In the following pages, all field names are those defined within that DSECT. The following pages also describe the various sections that might be present in the NEWS SMF record.

All records contain the header section. The header is followed by one section, or more, depending on whether the data is CNM record data or TARA statistical data.

SMF Header Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNMLEN	SMF record length	Binary
+2	+2	2	SMFNMSEG	Segment descriptor	Binary
+4	+4	1	SMFNMFLG	System indicator X‘3E’ for OS/390 or z/OS	Binary
+5	+5	1	SMFNMRTY	SMP record type, set by SYSPARMS SMFID=	Binary
+6	+6	4	SMFNMTME	Time stamp set by SMF in hundredths of a second	EBCDIC
+10	+A	4	SMFNMDTE	Record was moved to the external log buffer on this date. The format is 00YYDDDF where F is the sign	EBCDIC
+14	+E	4	SMFNMSID	System identifier	EBCDIC
+18	+12	1	SMFNMCAT	Record subcategory X‘03’ for CNM Deliver RU record X‘04’ for CNM record, not embedded	Binary
+19	+13	1	(Reserved)	X‘00’	Binary
+20	+14	12	SMFNMID	NetMaster system NMID value, set by SYSPARMS ID=	EBCDIC
+32	+20	40	(Reserved)	X‘00’	Binary
+72	+48	8	SMFNWNCP	Name of the NCP through which the device is connected. Blank, if the name is unknown.	EBCDIC
+80	+50	8	SMFNWLNK	Name of the link through which the device is connected. Blank, if the name is unknown.	EBCDIC
+88	+58	8	SMFNWPU	Name of the PU device. Blank, if the name is unknown.	EBCDIC
+96	+60	8	SMFNWLU	LU name, if applicable. Blank, if name is unknown.	EBCDIC

CNM Record Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+104	+68	Variable	SMFNWRU	The CNM record as it was received by Management Services	Binary

TARA Header Section

The header section for TARA data contains the following fields, in addition to those in the common SMF header section, described on page E-9.

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+104	+68	8	SMFWKSTA	Installation-defined string	EBCDIC
+112	+70	4	SMFWKSID	Workstation ID, WK*nn*, where *nn* is the workstation number	EBCDIC
+116	+74	Variable	SMFSTATS	Start of statistical information section	-

TARA Data Section

There can be one or more data sections for each SMF record in TARA data format. Each section has the following fields.

Offset Dec. where *n* is the start of the section	**Offset Hex.** where *n* is the start of the section	**Length Bytes**	**Field Name**	**Description**	**Type**
n+0	*n*+0	8	SMFTNAME	Installation-defined name to represent the type of information contained in this section	EBCDIC
n+8	*n*+8	3	SMFMIN	Minimum response time value	Binary
n+11	*n*+B	3	SMFMAX	Maximum response time value	Binary
n+14	*n*+E	4	SMFCUM	Total cumulative response time value	Binary
n+18	*n*+12	2	SMFINTV	Number of response time measurements	Binary
n+20	*n*+14	4	SMFRAVG	Average response time (that is, SMFCUM divided by SMFINTV)	Binary

F

Implementing the NTS User Exit

All session data that has been captured by NTS and queued for output processing is first passed to an installation-supplied user exit, where one exists. Following any exit processing, the session record is returned to NTS and considered for logging on the NTS database.

The user exit can perform any desired processing on the session data, and might indicate that the session record is to be ignored by the subsequent NTS logging function.

This appendix provides details of the parameters passed to the exit and the return codes required from the exit. Details of NTS SMF record formats are provided in Appendix G, *NTS SMF Record Formats*.

Sample NTS Exit

A sample NTS user exit named NTSXSMF is provided in source form and can be modified as required. The exit is a member of the *?dsnq*.SN400.SNSAMP library distributed on the installation tape.

Any DD cards required by the exit, if it is to use an external dataset, should be included in the Management Services execution JCL.

The sample exit is extensively documented and provides a (working) example of an exit that writes, to the System Management Facility (SMF) database, all NTS session data queued for output processing.

This exit takes the record as passed from NTS and inserts the SMF record type of 39. Since SMF fills out the SMF header area for system type records this is all that the exit need do before issuing the SMFWTM macro to write the record to SMF.

The Assembler macro $NMSMF, distributed in the Management Services macro library provides mapping for the format of the SMF record generated.

How the NTS Exit Is Called

Note

To implement an NTS exit, you must define the exit name in the SAW parameter group in ICS. See the *Unicenter NetMaster Network Management for SNA Implementation Guide*.

NTS is initialized when Management Services is started and a subtask that acts as the driver for the exit is attached. The subtask mainline routines handle communications between the subtask and the NTS components in the Management Services mother task.

When the subtask is attached, subtask mainline routines do the following:

Step 1. Load the load module specified as the installation-supplied user exit.

Step 2. Call the exit via conventional branching and linking.

Step 3. Pass an initialization parameter list to the exit.

Processing the NTS Exit

Because the user exit executes as part of a Management Services subtask, no restrictions are placed on the functions that the exit can perform. This is because the activities of the subtask do not impact the performance of the Management Services main task, and the exit subtask runs at a lower dispatching priority than the main task.

However, all processing of NTS session records on the output queue by the Management Services main task is delayed by processing occurring within the NTS user exit.

NTS Exit Coding Requirements

This section describes the coding requirements for the NTS exit.

Maintaining Registers on Entry to an Exit

You must observe standard linkage conventions on entry to an exit.

On entry, the registers must be saved in the caller’s save area, and on exit, restored (except R15, which contains a return code).

On entry, register contents are:

R1 Contains address of a parameter list.

R2-R12 Unpredictable.

R13 Contains address of caller’s standard save area.

R14 Caller’s return address.

R15 Contains address of entry point of user exit.

Parameter List Format

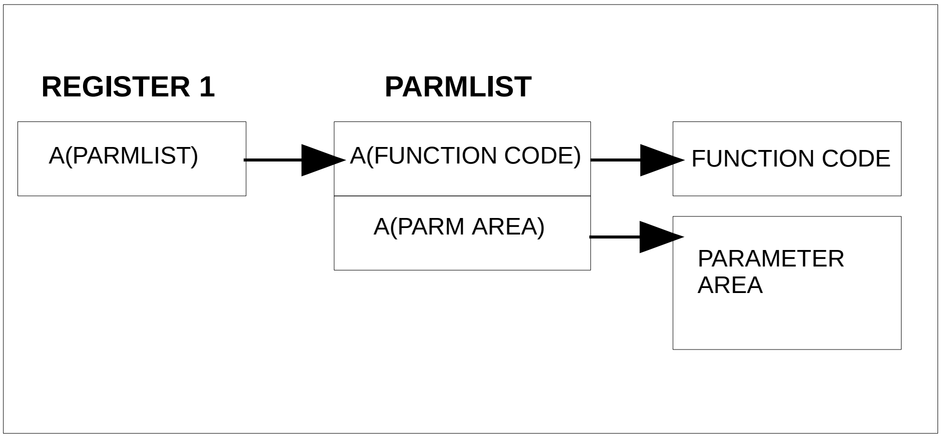
The parameter list addressed by Register 1 on entry consists of one or two contiguous fullwords. If there are two fullwords, the first contains the address of another fullword that holds a function code indicating the reason that the exit is being called.

The second fullword of the parameter list is present only for function codes 0 and 4. The high-order bit of this last (or solitary) fullword is not set, and the length of the parameter list must therefore be determined by examination of the function code.

- For function code 0, the second word of the parameter list contains the address of an area that contains system data that might be of value to the exit in determining its processing options.
- For function code 4, the second word of the parameter list contains the address of the NTS session record being passed to the exit for examination.
- For function code 8, the parameter list contains one word only.

The structure of the parameter list pointers therefore is as set out in Figure F-1.

Figure F-1. Structure of NTS User Exit Parameter List



About Exit Function Codes

The function code contained in a fullword addressed by the first word of the parameter list is a binary value that is right-aligned, with all high-order bits set to zero. Function codes used are:

- X‘00000000’ = initialization call
- X‘00000004’ = SMF record available
- X‘00000008’ = termination call

The processing associated with these function codes is detailed in this section.

Function Code 0

This indicates that the system has just been initialized. Any initialization processing that the exit needs to do, such as opening required datasets, should be done now.

The second word of the parameter list passed to the exit for function code 0, contains the address of an area, formatted as described next.

Bytes

00-03

In this fullword, the exit can store the address of a message that is to be logged to the Management Services activity log and sent to Monitor class operators. On return from any call to the exit, Management Services checks this word; if the value is non-zero, it is assumed to be the address of a message. (See the section, *Issuing Messages from the NTS Exit*, on page F-7 for the format of the messages.)

04-04

Operating system indicator. Value is:

X’10’

OS/390 or z/OS

05-05

SMF record identifier set by the SYSPARMS SMFID= command.

06-17

System identifier set by the SYSPARMS ID= command.

On successful completion of processing, the exit returns control to the caller with a completion code of 0 in register 15. Any other value in R15 is regarded as indicating that processing was unsuccessful and the exit subtask is terminated abnormally and assigned User Abend reason code 75D-01.

Function Code 4

This indicates that a session record has been placed on the NTS output queue. For this function code only, the second word of the parameter list contains the address of the record passed to the exit.

The session record passed to the exit is formatted as an SMF Type 39 system record. The full record layout is available in the macro DSECT $NMSMF, which is located in the distributed management services macro library.

To display the NTS session record description, enter:

```
label $NMSMF TYPE=NTS
```

Note that the area after the SMF record header contains variable information relating to the offset and length of those subsections present in the record. As the various data subsections are not always available to NTS, their inclusion is not guaranteed. This means that all access to such data subsections must proceed through the offset and length fields which relate to the subsections that are present. All offsets are from the first byte of the entire area passed (that is, the start of the SMF record header).

Record Subtype Identification

Only a copy of the session information is passed to the user exit and this data can be modified in any way without affecting subsequent NTS output processing. A halfword field labeled SMFNSUBT in the DSECT macro $NMSMF, and located at an offset of 22 bytes from the start of the record, contains the record subtype. NTS sets this field as follows:

01 The record passed contains RTM data collected for the session and was force-closed by the operator or closed during session awareness termination, but the session had not ended.

02 The record passed is a session end notification for a session that required NTS accounting.

03 The record passed is a session start notification for a session that requires the NTS accounting facility.

04 The record passed was force-closed by the operator, or closed during session awareness termination, but the session had not ended.

05 The record passed contains all data available at session end.

06 The record passed contains notification of a BIND rejection at session initialization.

07 The record passed contains notification of a session initialization failure that occurred before a BIND request was sent.

255 The record passed contains resource-based information. The type of information contained is indicated in the SMFNPSUB field. See *SYSPARMS Operands* in the *Management Services Administrator Guide*, for a detailed description of this field.

The processing performed by the exit on the record received is unrestricted, but it should be noted that exit processing is serialized and no additional NTS session records are processed on the output queue until the exit returns control to NTS.

Return Code Values

When control is returned by the exit, R15 must be set to one of the following (hexadecimal) return codes:

00 Processing complete. NTS continues processing the session record and passes the next record to the exit when it is available.

04 Processing complete. If this is a normal end of session record (record subtype 5) NTS does not log the record, otherwise this return code is treated in the same way as return code 0.

08	Processing complete. NTS processes the record but no further calls are made to the exit.
0C	Processing complete. As for return code 04, but no further calls are made to the exit.

Any other value in R15 is regarded as indicating unsuccessful processing and the exit subtask is abnormally terminated and assigned User Abend reason code 75D-02.

Function Code 8

This indicates that Management Services is terminating. It alerts the exit to perform any cleanup processing required, such as closing datasets. The termination of Management Services cannot proceed until the exit returns control.

On successful completion of processing, the exit returns control to the caller with a completion code of 0 in register 15. Any other value in R15 is regarded as indicating unsuccessful termination and the exit subtask is abnormally terminated and assigned User Abend reason code 75D-03.

Issuing Messages from the NTS Exit

You might require the exit to communicate with Management Services operators to notify them of particular conditions that have been detected.

You can use the exit to generate message text and place its address in a fullword contained in the area addressed by the second fullword of the initialization call parameter list. (For a description of the initialization call, see the section, *How the NTS Exit Is Called*, on page F-2.)

A message can be generated following any call to the exit, and its address placed in this fullword. The message must be formatted as follows:

Bytes

00-01	Length of message text (excluding these 2 bytes)
02-*nn*	Message text.

The maximum message length is 130 bytes. Excess length is ignored and the message truncated.

G

NTS SMF Record Formats

Session data and resource statistics captured by NTS are, when queued for output, first passed to an installation defined exit, if one exists. NTS organizes the data passed to the exit into records with a format compatible with that required by the System Management Facility (SMF) database.

The record is composed of a header, plus a number of other sections. Always included, and directly following the header, is the Data Section. It is used to indicate the presence of all other sections, and their location as an offset from the start of the entire SMF record. The presence of the optional sections will depend on the type of record being generated, and the information available to NTS at that time.

The macro $NMSMF is distributed with Management Services. It defines a DSECT describing the contents of the SMF records.

In the following pages all field names are those defined within that DSECT. Following is a description of the various sections that might be present in the NTS SMF record.

NTS SMF Record Description for All Sub-types

All records contain the following sections described below.

SMF Header Section

This section is present in all standard SMF records. NTS SMF records are recognizable by SMFNMRTY=X‘27‘ (SMF Type 39). Following the standard header is the Type 39 extension providing the product identifier and record sub-type field SMFNSUBT.

Data Section

This section is present in all NTS SMF records and provides a map giving the number and offsets for all other SMF data sections contained in the record.

Product Section

One product section is always present. This section includes the Management Services product identifier (NETM), the product version information, and the record sub-type field. This sub-type field, SMFNPSUB, is set to the same value as SMFNSUBT (above) except where SMFNSUBT=X’FF’ (Sub-type 255). Sub-type 255 records are NTS defined, and are further sub-divided by the SMFNPSUB field.

NTS SMF Record Sub-type 1 to 7 Description

Only session information records (Sub-types 1 to 7) contain the following sections described below.

Session Configuration Section

One session configuration section might be present, and includes:

- The type of session and session start and end times
- The names, types and positions in the network hierarchy of the session partners
- The MAI session user ID if the session is an MAI session and MAI session visibility is enabled

Session Accounting Section

One session accounting section might be present. It provides any accounting data collected for the session.

Session RTM Section

One session RTM section might be present. It provides any RTM data that might have been collected for the session.

Explicit Route Section

One explicit route data section might be present. It provides information about explicit routes associated with the subject session(s).

NTS SMF Record Sub-type 255 Description

Only resource statistics records (Sub-type 255) contain the following sections described below.

Resource Configuration Section

One resource configuration section might be present. It includes:

- The name, type and position in the network hierarchy of the resource
- Resource availability

Resource Accounting Section

One resource accounting section might be present. It provides any accounting statistics that might have been collected for the resource.

Resource RTM Data Section

One session RTM section might be present. It provides any RTM data that might have been collected for the resource.

SMF Header Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNMLEN	SMF record length	Binary
+2	+2	2	SMFNMSEG	Segment descriptor	Binary
+4	+4	1	SMFNMFLG	System indicator: X‘3E’ for OS/390 or z/OS	Binary
+5	+5	1	SMFNMRTY	Record type X‘27’	Binary
+6	+6	4	SMFNMTME	Time stamp set by SMF in hundredths of a second	EBCDIC
+10	+A	4	SMFNMDTE	Record was moved to the external log buffer on this date. The format is 00YYDDDF where F is the sign	EBCDIC
+14	+E	4	SMFNMSID	System identifier	EBCDIC
+18	+12	4	SMFNSID	NetMaster subsystem equals “NETM”	EBCDIC
+22	+16	2	SMFNSUBT	Record Sub-type number: X‘01’ for session RTM X‘02’ for session end X‘03’ for session start X‘04’ for session acct/avail X‘05’ for combined X‘06’ for BIND failure X‘07’ for INIT failure X‘FF’ for NTS data	Binary

Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	4	SMFNPOFF	Offset to product section	Binary
+4	+4	2	SMFNPLEN	Product section length	Binary
+6	+6	2	SMFNPNUM	Number of product sections	Binary
+8	+8	4	SMFNCOFF	Offset to configuration section	Binary
+12	+C	2	SMFNPLEN	Configuration section length	Binary
+14	+E	2	SMFNCNUM	Number of configuration sections	Binary
+16	+10	4	SMFNEOFF	Offset to explicit route data	Binary
+20	+14	2	SMFNELEN	ER data section length	Binary
+22	+16	2	SMFNENUM	Number of ER data sections	Binary
+24	+18	4	SMFNEOFF	Offset to TRM data section	Binary
+28	+1C	2	SMFNRLEN	RTM data section length	Binary
+30	+1C	2	SMFNRNUM	Number of RTM data sections	Binary
+32	+20	4	SMFNAOFF	Offset to accounting section	Binary
+36	+24	2	SMFNALEN	Accounting section length	Binary
+38	+26	2	SMFNANUM	Number of accounting sections	Binary

Product Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNPSUB	Record subtype for data same as SMFNSUBT except where SMFNSUBT= X‘FF’: X‘0001’forResource Statistics X‘0002’forResource Availability	Binary
+2	+2	2	SMFNPVER	Product version/ release equals X‘0032’ for MS V5.0.	Binary
+4	+4	4	SMFNPNAM	Product name equals “NETM”	EBCDIC

Session Configuration Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNCONR	Config data revision level equals X‘0032’ for MS V5.0.	Binary
+2	+2	8	SMFNCPLU	Primary resource name	EBCDIC
+10	+A	8	SMFNCPPU	Primary’s controlling PU	EBCDIC
+18	+12	8	SMFNCPLK	Primary’s controlling link	EBCDIC
+26	+1A	8	SMFNCPSU	Primary’s subarea PU	EBCDIC
+34	+22	8		Reserved	
+42	+2A	8	SMFNCSLU	Secondary resource name	EBCDIC
+50	+32	8	SMFNCSPU	Secondary’s controlling PU	EBCDIC
+58	+3A	8	SMFNCSLK	Secondary’s controlling link	EBCDIC
+66	+42	8	SMFNCSSU	Secondary’s subarea PU	EBCDIC
+74	+4A	8		Reserved	
+82	+52	8	SMFNCSCL	SAW class name for this session	EBCDIC
+90	+5A	8	SMFNCCOS	COS entry for this session	EBCDIC
+98	+62	2	SMFNCER	ER number for this session	Binary
+100	+64	2	SMFNCRER	Reverse ER number for session	Binary
+102	+66	2	SMFNCVR	VR number for this session	Binary
+104	+68	2	SMFNCTP	Trans pri for this session	Binary
+106	+6A	8	SMFNCCID	Unique VTAM session ID	Binary

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+114	+72	1	SMFNCSTY	Session Type: X‘01’ for LU/LU X‘02’ for SSCP/LU X‘03’ for SSCP/PU X‘04’ for SSCP/SSCP X‘05’ for LU-LU session through MAI X‘06’ for APPN CP-CP session	EBCDIC
+115	+73	1	SMFNCXNT	Cross network sess ind (Y/N)	EBCDIC
+116	+74	1	SMFNCUNB	BIND fail/UNBIND reason codes	Binary

Note

The following extension to the Session Configuration section is provided by NTS but is not usually found in the SMF Type 39 record.

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+117	+75	1	SMFNCATP	APPN transmission priority	Binary
+118	+76	2		Reserved	
+120	+78	8	SMFNCSTM	Session start time	Binary
+128	+80	8	SMFNCETM	Session end time Zero if session not ended.	Binary
+136	+88	8	SMFNCUSR	MAI session user ID Nulls if not an MAI session	EBCDIC
+144	+90	8	SMFNCACO	APPN class of service	EBCDIC
+152	+98	8	SMFCCPP	Control point name of APPN node which owns the PLU	EBCDIC
+160	+A0	8	SMFNCCPS	Control point name of APPN node which owns the SLU	EBCDIC

Note

All time stamps consist of the first 4 bytes of the system clock value, plus a 4-byte signed number being the time zone adjustment value in seconds.

Session Accounting Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNACCR	Accounting data revision level: X‘0032’ for MS V5.0.	Binary
+2	+2	2		Reserved	
+4	+4	8	SMFNASTM	Start time stamp Period start time for accounting data collection.	Binary
+12	+C	8	SMFNAETM	End time stamp Period end time for accounting data collection.	Binary
+20	+14	4	SMFNAPCP	Pri-Sec control PIUs	Binary
+24	+18	4	SMFNAPCB	Pri-Sec control bytes	Binary
+28	+1C	4	SMFNASCP	Sec-Pri control PIUs	Binary
+32	+20	4	SMFNASCP	Sec-Pri control bytes	Binary
+36	+24	4	SMFNAPTP	Pri-Sec text PIUs	Binary
+24	+18	4	SMFNAPTB	Pri-Sec text bytes	Binary
+28	+1C	4	SMFNASTB	Sec-Pri text PIUs	Binary
+20	+14	4	SMFNASTP	Sec-Pri text bytes	Binary

Note

All time stamps consist of the first 4 bytes of the system clock value, plus a 4-byte signed number being the time zone adjustment value in seconds.

Session Response Time Measurement Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNRTMR	RTM data revision level X‘0032’ for MS V5.0.	Binary
+2	+2	8	SMFNRSTM	Start time stamp Period start time for RTM data collection.	Binary
+10	+A	8	SMFNRETM	End time stamp Period start time for RTM data collection.	Binary
+18	+12	2	SMFNROPC	RTM objective percentage	Binary
+20	+14	2	SMFNROCT	RTM objective count	Binary
+22	+16	1	SMFNRDEF	RTM Definition	Binary
+23	+17	1	SMFNROOK	RTM Objective met? (Y or N)	EBCDIC
+24	+18	4	SMFNRTCT	RTM Total transaction count	Binary
+28	+1C	4	SMFNRTRT	RTM Total response time	Binary
+32	+20	4 X 4	SMFNRBND	RTM Boundary values	Binary
+48	+30	5 X 4	SMFNRCNT	RTM Boundary counts + Overflow	Binary
+68	+44	4	SMFNROT	RTM Objective Response Time	Binary

Note

All time stamps consist of the first 4 bytes of the system clock value, plus a 4-byte signed number being the time zone adjustment value in seconds.

Session Route Configuration Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNERR	Route element revision level	Binary
+2	+2	2	SMFNETOT	Route element total count	Binary
+4	+4	2	SMFNECNT	Route element present count There might be between 1 and 5 route elements present. Each route element entry occupies 10 bytes formatted as below.	Binary
+0	+0	8	SMFNESNM	Route element subarea name	EBCDI C
+8	+8	2	SMFNETG	Route element TG outbound	Binary

Resource Configuration Section

Offset Dec	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNRCR	Config data revision level: X‘0032’ for MS V5.0.	Binary
+2	+2	1	SMFNRCF1	Resource availability flag: ‘00’ for unavailable ‘80’ for available	Binary
+3	+3	1	SMFNRCTP	Resource Type: X‘F3’ for LU X‘F1’ for PU X‘FC’ for channel link X‘F9’ for TP link X‘F4’ for SSCP	EBCDIC
+4	+4	8	SMFNRCNW	Resource network ID	EBCDIC
+12	+C	8	SMFNRCNM	Resource name	EBCDIC
+20	+14	8	SMFNRCSS	Resource owning/adjacent SSCP	EBCDIC
+28	+1C	8	SMFNRCSP	Resource subarea PU name	EBCDIC
+36	+24	8	SMFNRCLN	Resource link name	EBCDIC
+44	+2C	8	SMFNRCPU	Resource PU name	EBCDIC
+52	+34	8		Reserved	
+60	+3C	8	SMFNRCST	Time of reported state change or end of interval time. That is, if SMFNPSUB=1 then this is the interval completion time; if SMFNPSUB=2 this is the time at which the named resource changed state.	Binary

Note

All time stamps consist of the first 4 bytes of the system clock value, plus a 4-byte signed number being the time zone adjustment value in seconds.

Resource Accounting Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNRAR	Accounting data revision level: X‘0032’ for MS V5.0	Binary
+2	+2	2		Reserved	
+4	+4	4	SMFNRAIN	Interval length (in seconds)	Binary
+8	+8	8	SMFNRAST	Interval start time stamp	Binary
+16	+10	8	SMFNRAET	Interval end time stamp	Binary
+24	+18	4	SMFNRASP	PIUs Sent	Binary
+28	+1C	4	SMFNRASB	Bytes sent	Binary
+32	+20	4	SMFNRASR	Response PIUs sent	Binary
+36	+24	4	SMFNRASC	Response byte count sent	Binary
+40	+28	4	SMFNRASN	Negative responses sent	Binary
+44	+2C	2	SMFNRASM	Maximum PIU data count sent	Binary
+46	+2E	2		Reserved	
+48	+30	4	SMFNRARP	PIUs received	Binary
+52	+34	4	SMFNRARB	Bytes received	Binary
+56	+38	4	SMFNRARR	Response PIUs received	Binary
+60	+3C	4	SMFNRARC	Response byte count received	Binary
+64	+40	4	SMFNRARN	Negative responses received	Binary
+68	+44	2	SMFNRARM	Maximum PIU data count received	Binary
+70	+46	2		Reserved	

Note

All time stamps consist of the first 4 bytes of the system clock value, plus a 4-byte signed number being the time zone adjustment value in seconds.

Resource Response Time Measurement Section

Offset Dec	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	SMFNRRR	RTM data revision level: X‘0032’ for MS V5.0	Binary
+2	+2	8	SMFNRRST	Interval start time stamp	Binary
+10	+A	8	SMFNRRET	Interval end time stamp	Binary
+18	+12	2	SMFNRROP	RTM objective percentage	Binary
+20	+14	2	SMFNRROC	RTM objective count	Binary
+22	+16	1	SMFNRRDF	RTM Definition	Binary
+23	+17	1	SMFNRROK	RTM Objective met? (Y or N)	EBCDIC
+24	+18	4	SMFNRRTR	RTM Total transaction count	Binary
+28	+1C	4	SMFNRRTM	RTM Total response time	Binary
+32	+20	4 X 4	SMFNRRBD	RTM Boundary values	Binary
+48	+30	5 X 4	SMFNRRCT	RTM Boundary counts + Overflow	Binary
+68	+44	4	SMFNRROB	RTM Objective Response Time	Binary
+72	+48	8	SMFNRRCL	RTM Class Name	EBCDIC

Note

All time stamps consist of the first 4 bytes of the system clock value, plus a 4-byte signed number being the time zone adjustment value in seconds.

H

NTS SNA Descriptor Table

SNA translation of hexadecimal codes to an equivalent SNA description for the NTS trace summary display is available to the installation through the modification of the supplied descriptor tables.

These tables are used for SNA code translation and provide:

- Meaningful descriptions associated with data flows
- New RUs and sense codes that you enter in the tables when defining them
- The ability to make installation-dependent and LU6.2 Function Management Headers (FMHs) visible on the NTS trace summary display

The member NMNTTABS distributed in the *?dsnq*.SN400.SNSAMP library contains the tables in their default form. Macros are provided such that new entries can be created as needed.

The table can be modified, compiled, and linked by the installation. Starting Session Awareness causes the tables to be reloaded. In this way the tables can be updated without the need to restart Management Services.

Macro Syntax

The three macros provided for the generation of table entries are:

$NTRUDEF

Used to define an RU description. It has syntax as described below:

```
$NTRUDEF CATEGORY=(FMD|DFC|NC|SC),
CODE=xxxxxxxx,
DESC='ccccccccccccccc'
```

where:

FMD is function management data.
DFC is data flow control.
NC is network control.
SC is session control.
xxxxxxxx is a hexadecimal string of up to 8 hex digits in length.
cccccccccccccc is a character string of up to 15 characters in length.

All definitions in the same category must be grouped together.

Example:

```
$NTRUDEF CATEGORY=SC,CODE=31,DESC='BIND'
```

$NTSCDEF

Used to define a sense code description. It has syntax as described below:

```
$NTSCDEF CATEGORY=(00|08|10|40|80),
SENSE=xx,
DESC='ccccccccccccccc'
```

where:

xx is a hexadecimal string two characters in length.
cccccccccccccc is a character string of up to 38 characters in length.

All definitions in the same category must be grouped together.

Definitions in the same category must be in ascending SENSE order.

Example:

```
$NTSCDEF CATEGORY=10,
SENSE=03,DESC='FUNCTION NOT SUPPORTED'
```

$NTFMHDF

Used to define a function management header description. It has syntax as described below:

```
$NTFMHDF FMH=xx,
DESC='cccccccccccccccc'
```

where:

xx is a hexadecimal string two characters in length.
cccccccccccccc is a character string of up to 15 characters in length.

Example:

```
$NTFMHDF FMH=12,DESC='FMH-12'
```

Table Formats

The member NMNTTABS in the *?dsnq*.SN400.SNSAMP library contains the tables in their default source form. All macros of a type are grouped together to form a table. In the case of the RU tables all RUs of the same category are grouped together to form sub-tables. Sense codes must be in ascending code sequence. Compile time error messages as described below are generated if these conditions are not adhered to.

Macro Compile Errors

Incorrect use of the macros provided generates the following errors:

$NTRUDEF

- MACRO CALLS OUT OF SEQUENCE
- INVALID CATEGORY SPECIFIED
- DESCRIPTION LENGTH EXCEEDS 15 CHARACTERS
- RUCODE LENGTH EXCEEDS 8 HEX DIGITS
- RUCODES CONSIST OF HEX DIGITS ONLY
- RUCODE APPEARS UNDER INCORRECT CATEGORY

$NTSCDEF

- MACRO CALLS OUT OF SEQUENCE
- INVALID CATEGORY SPECIFIED
- DESCRIPTION LENGTH EXCEEDS 38 CHARACTERS
- SENSE CODE LENGTH EXCEEDS 2 HEX DIGITS
- SENSE CODES CONSIST OF HEX DIGITS ONLY
- SENSE CODES MUST BE IN ASCENDING SEQUENCE

$NTFMHDF

- MACRO CALLS OUT OF SEQUENCE
- DESCRIPTION LENGTH EXCEEDS 15 CHARACTERS
- FMH CODE LENGTH EXCEEDS 2 HEX DIGITS
- FMH CODES CONSIST OF HEX DIGITS ONLY

Table Modification Procedure

The following steps should be completed to modify and implement changes to the table:

Note

All members are located in the *?dsnq*.SN400.SNSAMP dataset.

- Copy NMNTTABS to another dataset, for example the TESTEXEC library, and make the required modifications. Leave the original member intact in case of problems, and for any regular NetMaster for SNA maintenance.
- If you choose to compile and link NMNTTABS outside SMP, use the supplied sample JCL which is in the NTSASM member.
- In an SMP maintained system, however, you should compile a USERMOD using SMPCNTL and SMPPTFIN statements as shown in Figure H-1 below, and SMP receive and apply it. Then, if any maintenance is applied to the system which affects NMNTTABS, you are advised through the SMP Regression Report and you have to review your modifications and reapply the USERMOD.

  In the example illustrated in Figure H-1, the UCLIN update streams are adding the NetMaster for SNA SNMACROS to the SYSLIB concatenation, in order to make the NTS macros visible to the assembler. Once the UCLIN streams have been executed successfully, they can be deleted from the jobstream.
- Stop and restart session awareness by using the SAW parameter group in ICS.

Figure H-1. Example of SMP UCLIN Statements for a USERMOD

```
//SMP1.TESTEXEC DD DISP=SHR,DSN=?runq.TESTEXEC
//SMP1.SMPCNTL  DD *
  SET BDY(SOLVET1) .
  UCLIN .
  REP DDDEF(SYSLIB) CONCAT(SMPMTS,MACLIB,AMODGEN,SNMACROS) .
  ENDUCL .
  SET BDY(SOLVED1) .
  UCLIN .
  REP DDDEF(SYSLIB) CONCAT(SMPMTS,MACLIB,AMODGEN,SNMACLIB) .
  ENDUCL .
  SET BDY(GLOBAL) .
  RECEIVE S(NTSUMOD) .
  SET BDY(SOLVET1) OPTIONS(NMOPTNS) .
  APPLY S(NTSUMOD) REDO RC(RECEIVE=13) .
//SMP1.SMPPTFIN DD *
++USERMOD(NTSUMOD) .
++VER(Z038) FMID(NMD00NS)
++SRC(NMNTTABS) DISTLIB(SNSAMP) TXLIB(TESTEXEC) SYSLIB(SNLOAD)
                DISTMOD(SN1LOAD)
/*
```

I

NTS Storage Estimates

This appendix describes the storage estimates for active session data and history session data collected by NTS.

Active Session Data

All active session data which has been captured by NTS is kept in main memory. This storage is entirely above the 16 Mb line. The actual amount of storage required for any given network configuration will vary depending upon the NTS processing options. As a guide, the following scenario might be useful.

Scenario

Consider an NTS system operating in a single VTAM domain that contains 500 terminals, of which 400 are continually in session with one of several applications.

- If all active sessions (that is the 400 LU-LU sessions, 500 SSCP-LU sessions, plus a handful of SSCP-PU sessions) were to be kept by NTS, the storage requirement would be approximately 350K.
- By keeping only LU-LU sessions, this is reduced to about 180K.
- Gathering accounting data takes no extra storage.
- To collect RTM data for all 400 LU-LU sessions would require an extra 40K.

- If the default trace queue limits were used, and trace data collected for every LU-LU session, then this could reach a requirement of up to an extra 700K. However, it is not usual for all sessions to be concurrently holding the maximum number of trace PIUs in storage. New sessions take some time to reach this wrap level, while for ended sessions, the trace data is logged then purged from storage.

When Using NTS-SI

It is important to note that, if your NTS system is configured to receive session awareness data through NTS Single Image, then this must be considered when calculating the storage requirements of NTS. The local NTS receives session awareness from the remote NTS and maintain this data in storage. This data includes SSCP-SSCP, SSCP-PU, SSCP-LU and LU-LU session data. Session data (trace, RTM, and accounting) is solicited from the remote NTS when requested and is discarded when the data is not being viewed.

History Session Data

The space requirements of the NTS database vary according to the amount of data collected and logged. To store a single session incidence for a session name pair requires approximately 800 bytes. Each additional session incidence requires an extra amount of approximately 350 bytes. Further requirements are an extra 128 bytes for each record of session accounting or RTM data, and around 1200 bytes if trace data is logged (assuming the default trace queue depths are used).

Hence, for the default session keep count of 10 session incidence records per session name pair, the total database space requirement per session name pair is approximately:

- 4K when no additional session data is logged
- 6.6K when all accounting and RTM data is logged
- 18K when all trace data is additionally logged

Extrapolating further, with a 500-terminal network in which up to six different applications can be used by all terminals, this translates to an overall database storage requirement of approximately:

- 10 Mb when no additional session data is logged
- 18 Mb when all accounting and RTM data is logged
- 50 Mb when all trace data is additionally logged

Note

These figures are intended as a guide only for the initial implementation, as in operation many changing factors can affect the amount of data stored in the NTS database.

A Database Initialization Strategy

After you have installed and started using NTS, it might take a reasonably long period of time before all session name pairs have the number of session incidence records desired (that is, their desired session keep count values have been met). Up until this stage, new records are continually created and this causes the usual VSAM control interval splitting and its associated overheads. To assist in reducing such overheads, a possible strategy might be as follows:

1. Start session awareness with classes specifying logging on *all* (or, optionally, only the SSCP) sessions.
2. Ensure that all those resources that are usually going to partake in sessions, and are to be logged, are currently active.
3. Allow a short period for NTS to gather all current session status from VTAM before stopping session awareness with the CLOSE=ALL option to force log *every* session.

   Session awareness takes some time to terminate as it is busy logging sessions.

The object of this exercise is to collect on the NTS database at least one session incidence for every secondary resource. Having done this, you must reorganize the database. To do this:

Step 1. Start by copying the database to another dataset.

Step 2. Delete the NTS database.

Step 3. Reallocate the NTS database with very substantial free space requirements.

For example, to allow for more session incidences for each of the sessions captured, plus extra session name pairs for other applications that might be used by each terminal, perhaps up to 95 percent of the database should be allocated as free space. When the saved data is copied into the new database, this free space is available for insertions before any interval splitting need occur.

Glossary

This glossary defines the terms and abbreviations commonly used with Management Services.

It also includes references to terms used in an IBM environment and any equivalent FUJITSU terms.

3270 VDU terminal

An IBM video display terminal. This is often used to refer to the entire range of 3270 terminals. When followed by a number (for example, 3270-5), a specific model is intended.

370/390

This is an abbreviation for IBM’s System 370 or S/390 architecture. It is often used to indicate any mainframe CPU that implements this architecture.

3745/3746-900

An IBM front end communications processor. (The Fujitsu equivalent is a CCP or 2806.)

ACB

See *Access method Control Block*

Access Method Control Block (ACB)

A control block that links an application program to an access method such as IBM’s VTAM or VSAM.

Access Security Exit

An installation-provided routine that may be used to replace the Management Services UAMS functions, partially or completely, allowing logon, logoff, and password maintenance requests to be passed to an external security system.

ACF/NCP

Advanced Communications Function for the Network Control Program. Synonym for *NCP*.

ACF/VTAM (Advanced Communication Facility/VTAM)

IBM’s product implementation of SNA’s SSCP or CP.

Active Link

A link that is currently available for transmission of data.

Activity Log

A system-maintained log that records all important activity for use in later problem determination.

Adapter

A part that electrically or physically connects a device to a computer or to another device.

Adjacent Control Point

A Control Point (CP) that is directly connected to an APPN, LEN, or composite node by a link.

Adjacent Link Station (ALS)

(1) In SNA, a link station directly connected to a given node by a link connection over which network traffic can be carried.

Note

Several secondary link stations that share a link connection do not exchange data with each other and therefore are not adjacent to each other.

(2) With respect to a specific node, a link station partner in an adjacent node.

Advanced Peer-to-Peer Networking (APPN)

An extension to SNA featuring:

- Greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure
- Dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection
- Dynamic definition of network resources
- Automated resource registration and directory lookup

APPN extends the LU6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

Advanced Program to Program Communications (APPC)

An IBM-defined application level protocol which makes use of SNA's LU 6.2.

Alert

(1) A message sent to a Management Services focal point in a network to identify a problem or an impending problem.
(2) In SNA Management Services (SNA/MS), a high priority event that warrants immediate action.

ALS

See *Adjacent Link Station.*

ANR

See *Automatic Network Routing.*

APF

See *Authorized Program Facility.*

APPC

See *Advanced Program to Program Communications.*

APPL

A VTAM term used to describe the definition that allows an application to use VTAM facilities.

APPN

See *Advanced Peer-to-Peer Networking.*

APPN Network

A collection of interconnected network nodes and their client nodes.

APPN Network Node

A node that offers a broad range of end-user services and that can provide the following:

- Distributed directory services, including registration of its domain resources to a central directory server
- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service.
- Session services for its local LUs, and client end nodes
- Intermediate routing services within an APPN network

Automatic Network Routing (ANR)

In high performance routing, a highly efficient routing protocol that minimizes cycles and storage requirements for routing network layer packets through intermediate nodes on the route.

ASN.1

Abstract Syntax Notation One, defined by ISO 8824, is an abstract syntax used to describe data structures. It is used by Mapping Services to define data structures within Management Services.

Authorized Program Facility (APF)

Describes the special authorization level required within the operating system for certain applications.

Backup Focal Point

A focal point that provides Management Services support for a particular category for a node in the event of a communications failure with the primary focal point. Both assigned focal points (explicit and implicit) and default focal points can have backup counterparts. Contrast with *Primary Focal Point.*

Backward Explicit Congestion Notification (BECN)

A bit set by a frame relay network to notify an interface device that congestion avoidance procedures should be initiated by the sending device.

Beaconing

Pertaining to repeated transmission of a beacon message when a normal signal is not received because of a serious fault, such as a line break or power failure. The message is repeated until the error is corrected or bypassed.

BECN

See *Backward Explicit Congestion Notification.*

BIND

(1) A VTAM term describing the action of logically linking one network resource with another network resource.
(2) In SNA, a request to activate a session between two logical units.

Border Node

An APPN network node that interconnects APPN networks having independent topology databases in order to support LU-LU sessions between these networks.

Boundary Node

In SNA, a subarea node with boundary function.

> **Note**
>
> A subarea node may be a boundary node, an intermediate routing node, both, or neither, depending on how it is used in the network.

Broadcast Services

Broadcast Services controls the sending of messages throughout NetMaster systems.

CDRM

See *Cross-Domain Resource Manager.*

CDRSC

See *Cross-Domain Resource*.

Central Directory Server

A network node that provides a repository for information on network resource locations; it also reduces the number of network searches by providing a focal point for queries and broadcast searches and by caching the results of network searches to avoid later broadcasts for the same information.

Channel Adapter

A communication controller hardware unit that is used to attach the communication controller to a host channel.

Checkpoint

Refers to a point of synchronization in processing where a unit of work is complete, or partially complete, such as where data is recorded for restart purposes. A point at which information about the status of transmission can be recorded so that it can be restarted later.

Class of Service (CoS)

A set of characteristics (such as route security, transmission priority, and bandwidth) used to construct a route between session partners. The CoS is derived from a mode name specified by the initiator of a session.

Client

A functional unit that receives shared services from a server.

CNM

See *Communications Network Management*.

CNMPROC

The name given to an NCL procedure used to intercept CNM records received across the VTAM CNM interface by the NEWS component of NetMaster for SNA.

Command Partition

A term associated with NPF that describes the group of network resources a user ID is authorized to reference with VTAM commands.

Communications Network Management (CNM)

IBM term for its SNA management facilities.

Configuration Management

An ISO/OSI classification of management functions that apply to the ability to set or change operating parameters of the system, to collect and distribute information on their status, to associate names with the entities, and to change the system configuration.

Congestion

See *Network Congestion.*

Control Member

A term associated with NPF that describes the list of resource table names applying to a user ID. This control member is referenced in the definition of USERID.

Control Point

(1) A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network.
(2) A component of a node that manages resources of that node and optionally provides services to other nodes in the network. Examples are a System Services Control Point (SSCP) in a type 5 subarea node, a network node control point in an APPN network node, and an end node control point in an APPN or LEN end node.

CP-CP Sessions

The parallel sessions between two control points, using LU 6.2 protocols and a mode name of CPSVCMG, on which network services requests and replies are exchanged. Each CP of a given pair has one contention-winner session and one contention-loser session with the other.

CP Name

A network-qualified name of a Control Point (CP), consisting of a network ID qualifier identifying the network (or name space) to which the CP's node belongs, and a unique name within the scope of that network ID identifying the CP.

Cross Domain

In SNA, pertaining to control or resources involving more than one domain.

Cross-Domain Resource

A VTAM term describing the definition of a network resource that is owned by a VTAM in another domain.

Cross-Domain Resource Manager (CDRM)

In VTAM, the function in the System Services Control Point (SSCP) that controls initiation and termination of cross-domain sessions.

Data Link Connection Identifier (DLCI)

The numeric identifier of a frame relay subport of a PVC segment in a frame relay network. Each subport in a single frame relay port has a unique DLCI.

DLCI

See *Data Link Connection Identifier.*

Domain

1. An SNA term describing a domain that consists of the set of SNA resources controlled by one common control point called an SSCP. In terms of implementation, an SSCP is the host access method (VTAM). An SNA network consists of one or more domains.
2. A VTAM term that describes a logical division of a network. Networks are divided into domains that are associated with the way they are controlled.

Domain ID

Term for a 1-4 character mnemonic used as a unique identifier for a NetMaster system.

Dynamic Allocation

Assignment of datasets to a program at the time the program is executed rather than at the time the job is started.

End Node

In communications, a node that is frequently attached to a single data link and cannot perform intermediate routing functions.

ER

See *Explicit Route.*

Exception

The result of a service request that did not complete successfully. See *Reply* and *Response*.

Exit

An installation-written routine that can be driven from a point within a program to provide data to the program, or perform additional processing relevant to that installation's specific requirements.

Explicit Route (ER)

In SNA, a series of one or more transmission groups that connect two subarea nodes. An Explicit Route is identified by an origin subarea address, a destination subarea address, an Explicit Route number, and a reverse Explicit Route number. Contrast with *Virtual Route.*

Extended Datastream

A 3270 datastream containing fields that utilize color and extended highlighting capabilities of the terminal.

Extended Multiple Console Support (EXTMCS)

EXTMCS consoles are consoles that the SYSCMD facility can use as an alternative to JES consoles in an MVS/ESA 4.1, or later, environment.

EXTMCS

See *Extended Multiple Console Support*.

FCS

See *Finance Communications System*.

FECN

See *Forward Explicit Congestion Notification.*

Finance Communications System (FCS)

A system used by banks and other large financial institutions.

Forward Explicit Congestion Notification (FECN)

A bit set by a frame relay network to notify an interface device that congestion avoidance procedures should be initiated by the receiving device. Contrast with *BECN.*

Frame Handler

Synonym for *Frame Relay Frame Handler (FRFH).*

Frame Handler Subport (FHSP)

The access point of a frame relay frame handler to a PVC segment. Frame Handler SubPorts function in pairs; frames enter the frame handler through one frame handler subport and exit through the other. Contrast with *Terminating Equipment Subport*.

FHSP

See *Frame Handler Subport*.

Frame Relay

(1) An interface standard describing the boundary between a user's equipment and a fast-packet network. In frame relay systems, flawed frames are discarded; recovery comes end-to-end rather than hop-by-hop.
(2) A technique derived from the integrated services digital network (ISDN) D channel standard. It assumes that connections are reliable and dispenses with the overhead of error detection and control within the network.

Frame Relay Connection

See *Frame Relay Physical Line* and *Permanent Virtual Circuit (PVC).*

Frame Relay Frame

The frame relay frame structure defined by American National Standards Institute (ANSI) Standard T1.618.

Frame Relay Frame Handler (FRFH)

(1) The function in a frame relay node that routes (or switches) frames along a permanent virtual circuit. A frame handler receives frames from an adjacent frame relay node and uses the DLCI to forward them to the next node on the PVC. Synonymous with frame handler. See also *frame relay switching equipment support* and *frame relay terminating equipment*.
(2) In NCP, the function that switches frames between frame handler subports on an internal PVC segment. The NCP frame handler function can also switch frames to the frame relay terminating equipment function.

Frame Relay Network

A network that consists of frame relay handlers and in which frames are passed from one frame relay terminating equipment station to another through a series of one or more frame relay frame handlers.

Frame Relay Physical Line

The physical connection between two frame relay nodes. A frame relay physical line can simultaneously support PVC segments for both the frame handler and terminating equipment functions. In NCP, a frame relay physical line is defined as a nonswitched duplex line.

Frame Relay Switching Equipment (FRSE)

See *Frame Relay Switching Equipment Support.*

Frame Relay Switching Equipment Subport Set

The set of primary and, optionally, substitute frame handler subports within an NCP that comprise those used for a given frame relay segment set.

Frame Relay Switching Equipment Support

In NCP, a set of frame relay functions that include the frame relay frame handler function and the Local Management Interface (LMI) function. These functions are defined by American National Standards Institute (ANSI) Standards T1.617 and T1.618, and International Telegraph and Telephone Consultative Committee (CCITT) Standards Q.922 and Q.933. NCP provides additional functions, including performance measurement and enhanced reliability, that are not defined by ANSI or CCITT standards.

Frame Relay Terminal Equipment (FRTE)

A device that can connect to a frame relay network and provide the Frame Relay Terminating Equipment function. See also *Frame Relay Frame Handler* and *Frame Relay Terminating Equipment.*

Frame Relay Terminating Equipment

The function at the end of a frame relay permanent virtual circuit. Frame relay terminating equipment provides higher-layer protocols with access to a frame relay network through terminating equipment subports. It does by (a) adding frame relay frame headers to data for another protocol and sending the frames to adjacent frame relay nodes, and (b) receiving frames from adjacent frame relay nodes and removing the frame headers. See also *Frame Relay Frame Handler*, *Frame Relay Switching Equipment Support*, and *Frame Relay Terminal Equipment.*

Frame Switching

The function performed by frame relay nodes to route frames through a network. See also *Frame Relay Frame Handler.*

FRFH

See *Frame Relay Frame Handler.*

FRSE

See *Frame Relay Switching Equipment.*

FRTE

See *Frame Relay Terminating Equipment.*

Gateway

(1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks of system with the same or similar architectures.
(2) The combination of machines and programs that provide address translation, name translation, and System Services Control Point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway VTAM.
(3) In the IBM token-ring network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols.
(4) In TCP/IP, synonym for *router.*
(5) To the routing layer, the logical distance between two nodes in a network.

ICS (Initialization and Customization Services)

Initialization and Customization Services is an AutoAssist facility that helps you set up your region parameters.

IMS

See *Information Management System*.

Information Management System (IMS)

IBM’s database/data communication (DB/DC) system that can manage complex databases and networks.

INMC

See *Inter-Management Services Connection.*

Inter-Management Services Connection (INMC)

This facility allows systems running in a network containing multiple CPUs to communicate with each other, providing general-purpose data transfer between CPUs within the network. INMC provides the capability for up to sixteen sessions between any pair of systems. In appropriate systems, these sessions can traverse different physical network paths, thus increasing throughput. This component also provides additional link security and management facilities.

Inter-System Routing (ISR)

ISR is used to propagate system and network management information between systems in the network.

Interchange Node

A VTAM node that acts as both an APPN network node and a type 5 subarea node to transform APPN protocols to subarea protocols and vice versa.

Intermediate Network Node

(1) In APPN, a node that is part of a route between an Origin Logical Unit (OLU) and a Destination Logical Unit (DLU), but does not contain the OLU or DLU and does not serve as the network server for the OLU or DLU.
(2) In VTAM, deprecated term for *intermediate routing node.*
(3) In NCP, deprecated term for *subarea node.*

ISR

See *Inter-System Routing*.

JES (Job Entry Subsystem) Consoles

Virtual consoles, defined when you first perform the NetMaster for SNA initial startup procedure. These consoles can be acquired by an authorized program for use in issuing MVS and subsystem commands.

Key Sequenced Data Set (KSDS)

A VSAM dataset whose records are directly accessed by a user-supplied key.

KSDS

See *Key Sequenced Data Set.*

LAN

See *Local Area Network.*

LEN

See *Low Entry Networking.*

Link

A term used to describe a logical connection between two or more systems. See also *INMC (Inter-Management Services Connection)*.

Link Service Access Point

In the IBM token-ring network, the logical point at which an entity in the Logical Link Control (LLC) sublayer provides services to the next higher layer.

LLC

See *Logical Link Control.*

LMI

See *Local Management Interface Protocol.*

LMI Subport

A frame relay subport that exchanges line status information with adjacent nodes using Local Management Interface (LMI) protocol. In NCP, the LMI subport is the link-station subport for the physical line.

Local Area Network (LAN)

(1) A computer network located on a user’s premises within a limited geographical area. Communication within a LAN is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation.
(2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.
Contrast with *WAN.*

Local Management Interface (LMI) Protocol

In NCP, a set of frame-relay network management procedures and messages used by adjacent frame-relay nodes to exchange line status information over DLCI X‘00’. NCP supports both the ANSI and CCITT versions of LMI protocol. These standards refer to LMI protocol as link integrity verification tests.

Logical Line

The representation of the connection between NCP and a node communicating with NCP over a physical line such as token ring or frame relay. A single physical line can support multiple logical lines. Contrast with *physical line.*

Logical Link Control (LLC)

The protocol in a LAN that governs the exchange of transmission frames between data stations regardless of how the transmission medium is shared

Logical Unit (LU)

The point of access for any user to an SNA network. SNA introduced the concept of the LU. The LU is a type of SNA Network Accessible Unit (NAU) that provides protocols for end users to gain access to the network and to the functional components of the LUs.

LOGMODE

A VTAM term used to describe a table entry that defines the characteristics and protocols of a terminal.

Logoff

A request by an LU that it be disconnected from a VTAM application program.

Logon

A request by an LU that it be connected to a VTAM application program.

LOGPROC

The name given to an NCL procedure used to intercept messages destined for the Management Services activity log.

Low Entry Networking (LEN)

A capability of nodes to attach directly to one another using basic peer-to-peer protocols to support multiple and parallel sessions between LUs.

LSR (Local Shared Resources)

A technique for buffering I/O to VSAM files called LSR pools. NCL supports this type of processing for User Databases (UDBs).

LU

See *Logical Unit*.

LU0

An unconstrained SNA protocol that allows implementers to select any set of available protocol rules, as long as the two LUs are able to communicate with each other successfully according to the rules chosen. Therefore, all LU types are an implementation of LU Type 0.

LU1

A line-by-line or typewriter type terminal (for example 3767, 3770), using SNA protocols.

LU2

A 3270 type terminal using SNA protocols.

LU3

LU Type 3 was implemented to support printers with a different data stream format. LU Type 3 is used by printers attached to an IBM display cluster controller.

LU4

LU Type 4 was implemented so that office system products could transfer documents. LU Type 4 is used by banking devices.

LU6.2

A protocol that serves as a port into an SNA network. LU6.2 defines a specific set of services, protocols, and formats for communication between logical processors. LU6.2 provides presentation services for presentation of data to the end user, transaction services for performing transaction processing on behalf of the end user and LU services for managing the resources of the LU.

LU7

An SNA protocol that is used by word-processing devices.

MAC

See *Medium Access Control.*

Major Node

In VTAM, a set of resources that can be activated and deactivated as a group. Contrast with *Minor Node.*

Management Services

Management Services provides the central core of functions and service routines for the 3270 Unicenter, NetMaster, and SOLVE products.

Management Services Unit (MSU)

A data unit in an SNA network. There are various types of SNA MSUs, and many reasons for the generation of each type of MSU.

Mapped Data Object (MDO)

Any data item that can be represented as a continuous string a bytes in storage.

MDO

See *Mapped Data Object.*

Medium Access Control (MAC)

In LANs, the sublayer of the data link control layer that supports medium-dependent functions and uses the services of the physical layer to provide services to the logical link control sublayer. The MAC sublayer includes the method of determining when a device has access to the transmission medium.

Message Partition

A term associated with NPF that describes the group of network resources for which a user ID will receive unsolicited (PPO) VTAM messages.

Minor Node

In VTAM, a uniquely defined resource within a major mode. See *Major Node* and *Node.*

MSGPROC

An NCL procedure used to intercept and process messages destined for a user's Operator Console Services (OCS) window.

MSU

See *Management Services Unit.*

NAU

See *Network Accessible Unit (NAU).*

NCL

See *Network Control Language.*

NCL Procedure

A member of the procedures dataset comprising NCL statements and Management Services (MS) or VTAM commands. The NCL statements and other commands are executed from an EXEC or START command specifying the name of the procedure.

NCL Process

The NCL task that is invoked, usually by a START command to execute one or more associated procedures. Each NCL process has a unique NCL process identifier.

NCL Processing Environment

Provides the internal services and facilities required to execute NCL processes for the user, from its associated window.

NCL Processing Region

All users (real or virtual) have an NCL Processing Region associated with their user ID while logged on. This region provides all of the internal services needed to allow the user to have processes executed on their behalf. There may be a maximum of two active NCL environments in a user's NCL region.

NCLID

A 6-digit NCL process identifier which is unique within the system. It is used to identify a process for the purpose of communicating with that process.

NCP

See *Network Control Program.*

NCS

See *Network Control Services*.

NDB

The NetMaster Database. An extension to NCL which provides a relational database facility that can be used as a repository for applications running within a system. Full update capabilities, including scans with extensive Boolean logic, are provided.

Network Accessible Unit (NAU)

In SNA, a logical unit, a physical unit, or a system services control point. The NAU is the origin or destination of information transmitted by the path control network. Synonymous with network accessible unit.

Network Control Language (NCL)

The interpretive language that allows logical procedures (programs) to be developed externally to Management Services and then executed by Management Services on command. NCL contains a wide range of logic, built-in functions and arithmetic facilities which can be used to provide powerful monitoring and automatic control functions.

Network Control Program (NCP)

This resides within and controls the operation of a communications controller. The NCP communicates with VTAM.

Network Control Services (NCS)

A facility of NetMaster for SNA that provides full screen displays and navigation of the network in Management Services.

Network Error Warning System (NEWS)

A facility of NetMaster for SNA which is used to provide network error and traffic statistics and error alert messages.

Network Management Vector Transport (NMVT)

A Management Services Request/response Unit (RU) that flows over an active session between Physical Unit management services and Control Point Management Services (SSCP-PU session).

Network Node

See *APPN Network Node.*

Network Partitioning Facility (NPF)

A facility of Management Services that allows the range of resources which an operator can influence to be denied.

Network Tracking System (NTS)

A facility of NetMaster for SNA used to provide SNA session monitoring, dynamic online network tracing, accounting, and response time information in conjunction with diagrammatic representations of session partners.

NEWS

See *Network Error Warning System.*

NMINIT

The NCL procedure automatically executed after system initialization has completed. It cannot contain commands that require VTAM facilities as it is executed before the primary ACB is opened. The procedure name can be changed by the installation.

NMREADY

The NCL procedure automatically executed once system initialization has completed. It can contain commands that require VTAM facilities as it is executed after the primary ACB is opened. Procedure name can be changed by the installation.

NMVT

See *Network Management Vector Transport*.

Node

(1) A connection point in a communications network.
(2) In network topology, the point at an end of a branch.
Any device, attached to a network that transmit and receives data.
(3) An endpoint of a link or a junction common to two or more links in a network. Nodes can be processors, communication controllers, cluster controllers, or terminals. Nodes can vary on routing and other functional capabilities,
(4) In VTAM, a point in a network defined by a symbolic name.
See *Major Node* and *Minor Node.*

NPF

See *Network Partitioning Facility.*

NPF Control Member

A member of the NPF dataset member which defines a list of member names that are to be the resource tables for the associated user ID.

NPF Resource Table

A member of the NPF dataset that defines a group of network resource names. The resource names can be defined specifically or generically using wildcard characters. A resource table is pointed to by a control member.

NT 2.1

Node Type 2.1. A node in an SNA network. It implements a peer-to-peer protocol and allows greater dynamics in network configuration, greater independence in session set up between partner LUs and reduced definitions.

NTS

See *Network Tracking System.*

OCS

See *Operator Console Services*.

Operator Console Services (OCS)

A facility of Management Services that provides general operational control and an advanced operator interface to VTAM for network management.

OS/390

An IBM operating system.

Packet

The logical unit of transmission in a network.

Partitioned DataSet (PDS)

A type of dataset format that supports multiple individual members in the one physical dataset. Equivalent to the Source Statement Library in SP systems.

Path Information Unit (PIU)

An SNA packet.

PDS

See *Partitioned DataSet*.

Peer

In network architecture, any functional unit that is in the same layer as another entity.

Peer-to-Peer Management

A non-hierarchical heterogeneous network management system.

Permanent Virtual Circuit (PVC)

(1) In X.25 and frame relay communications, a virtual circuit that has a logical channel permanently assigned to it at each data terminating equipment (DTE). Call-establishment protocols are not required. Contrast with switched virtual circuit (SVC).
(2) The logical connection between two frame relay terminating equipment stations, either directly or through one or more frame relay frame handlers. A PVC consists of one or more PVC segments.

Physical Line

The physical connection between NCP and an adjacent device or local area network (LAN). A single physical line, such as token ring or frame relay, can support multiple logical lines. Contrast with *logical line*.

Physical Unit (PU)

(1) The control unit or cluster controller of an SNA terminal. The part of a control unit or cluster controller which fulfils the role of an SNA-defined PU.
(2) Each node (a logical grouping of hardware) in an SNA network is addressed by its PU. There are 4 types of nodes or PU in an SNA network: PU-T5, PU-T4, PU-T2, PU-T1. See *PU Type* x. A PU is a type of NAU. Contrast with *Network Accessible Unit (NAU)*.

PIU

See *Path Information Unit.*

PLU

See *Primary Logical Unit.*

PPI

See *Program to Program Interface.*

PPO

See *Primary Program Operator.*

PPOPROC

The name given to the NCL procedure used to intercept unsolicited VTAM (PPO) messages .

Primary and Secondary

(SNA) Primary and secondary are SNA terms for describing the LU's role when the session is established. The primary LU sends the BIND request that causes the session to be established, and the secondary LU receives the BIND request. Rules defined in the BIND request determine which of these is the first speaker in the exchange of information.

Primary Focal Point

A focal point understood to be the preferred source of Management Services support for a particular category. Contrast with *Backup Focal Point.*

Primary Logical Unit (PLU)

In SNA, a type of LU that is usually used by the application programs in a host. It refers to the BIND sender for a session.

Primary Program Operator (PPO)

A VTAM term that describes a facility of VTAM that allows unsolicited network messages to be delivered to an application program, such as Management Services, for processing. (See also **SPO**.)

Primary Route

In NCP frame relay, the internal PVC segment between the two primary frame handler subports in a subport set. Contrast with *substitute route.*

Program to Program Interface (PPI)

PPI is a general-purpose facility which allows programs, written in any language, to exchange data.

PU

See *Physical Unit*.

PU Type 1

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a terminal (such as an IBM 3278).

PU Type 2

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a cluster controller (such as an IBM3*x*74, 3276, 3770 or 3790).

PU Type 4

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a communications controller (such as an IBM 3704, 3705, 3725 or 3745).

PU Type 5

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a host computer system (such as an S/390 or z900, running VTAM or sometimes VCAM).

PVC

See *Permanent Virtual Circuit.*

Quiesce

(1) To end a process by allowing operations to complete normally
(2) To request that a node stop sending synchronous-flow messages

Quiesce Protocol

In VTAM, a method of communicating in one direction at a time. Either the primary logical unit or the secondary logical unit assumes the exclusive right to send normal-flow requests, and the other node does not send such requests. When the sender wants to receive, it releases the other node from its quiesced state.

Rapid Transport Protocol (RTP) Connection

In high-powered routing, the connection established between the endpoints of the route to transport session traffic.

RECFMS

See *Record Formatted Maintenance Statistics*.

RECMS

See *Record Maintenance Statistics*.

Record Formatted Maintenance Statistics (RECFMS)

A statistical record built by an SNA controller and usually solicited by the host.

Record Maintenance Statistics (RECMS)

An SNA error event record built from an NCP or line error and sent unsolicited to the host.

Remote Operator Facility (ROF)

A facility of Management Services that allows an operator to sign on to a remote location, execute commands and have the results returned.

Reply

The information returned to a directive as a result of a request. This information may be either a response or an exception, together with appropriate arguments. See *Exception* and *Response*.

REQMS

See *Request for Maintenance Statistics.*

Request

The invocation of a directive, together with appropriate arguments. See *Exception* and *Response*.

Request for Maintenance Statistics (REQMS)

A host solicitation to an SNA controller for a statistical data record.

Request Unit (RU)

(SNA) A message unit that contains control information such as a request code, or function management headers, end-user data, or both.

Resource Table

A term associated with NPF that describes a list of resource names or generic resource names that define a command or message partition .

Response

The success result of a service request. See *Exception* and *Reply.*

Response Time Measurement (RTM)

Measurement of the time which passes between the user starting an action (by pressing a key) and the response appearing on the screen.

Response Time Monitor (RTM)

A facility provided by IBM's 3*x*74 control units to monitor end-user response times. NEWS can interpret this data .

Response Unit (RU)

(SNA) A message unit that acknowledges a request unit.

Return Code

A code returned from the system that indicates the success or failure of the task performed.

ROF

See *Remote Operator Facility.*

Route

(1) An ordered sequence of nodes and Transmission Groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them.

(2) The path that network traffic uses to get from source to destination.

Router

(1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses.
(2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer.
(3) In OSI terminology, a function that determines a path by which an entity can be reached.
(4) In TCP/IP, synonymous with *gateway.*
(5) Contrast with *bridge.*

RouTing update Protocol (RTP)

The VIrtual NEtworking System (VINES) protocol that maintains the routing database and allows the exchange of routing information between VINES nodes.

RTM

(1) See *Response Time Measurement.*
(2) See *Response Time Monitor.*

RTP

See *Rapid Transport Protocol Connection.*

RU

(1) See *Request Unit.*
(2) See *Response Unit.*

SAW

See *Session Awareness Data.*

SDLC

See *Synchronous Data Link Control.*

Secondary Logical Unit (SLU)

In SNA, a type of LU that is usually used by the end-users at the terminals or by programs which reside in the peripheral node.

Secondary Program Operator (SPO)

A VTAM term that describes a facility of VTAM that allows only messages generated by commands issued by an application program, such as Management Services, to be delivered to the application program for processing. Unsolicited messages are not delivered. Contrast with *PPO.*

Security Initialization Unit

A hardware device that creates and loads encrypting codes, also known as *keys*, for your computer system.

Sequence Number

A number assigned to each message exchanged between a VTAM application program and an LU. Values increase by one throughout the session, unless reset by the application program using an STSN or CLEAR command .

Server

A process designed to serve the data to a client, or request process, for one or more users.

Service Point

An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

Session

(1) In network architecture, for the purpose of data communication between functional units, all the activities which take place during the establishment, maintenance, and release of the connection.

(2) A logical connection between two Network Accessible Units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header accompanying any transmissions exchanged during the session.

Session Awareness Data (SAW)

A type of network management data supplied by VTAM and processed by NTS.

Session Name

A name assigned to a workstation or session to permit it to receive messages or share resources.

Session Replay Facility (SRF)

A part of the MAI/EF facility of SOLVE:Access which provides the ability to record and playback terminal session scenarios .

SIS

See *Screen Image Services.*

SLU

See *Secondary Logical Unit.*

SMF

See *System Management Facility.*

SNA

See *Systems Network Architecture.*

SOLVE

The term SOLVE encompasses the services provided by Management Services and Automation Services. For example, the SOLVE PPI is a service provided by the subsystem interface (SSI) in Management Services.

SPO

See also *Secondary Program Operator.*

SRF

See *Session Replay Facility.*

SSCP

See *System Services Control Point.*

Structured field

Representation of user ID attribute information exchanged between Management Services and its security exit.

Subarea

A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all Network Accessible Units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

Subport

(1) An access point for data entry or exit over a logical connection. The relationship between the physical line and the port is analogous to the relationship between the logical connection and the subport. (2) In a frame relay network, the representation of a logical connection on a frame relay physical line and the point where the logical connection attaches to the frame relay frame handler. Each subport on a physical line has a unique data link connection identifier and can represent an FRTE, FRFH, or LMI connection. See *Frame Handler Subport* and *Terminal Equipment Subport.*

Subport Set

In NCP, a set of frame handler subports linked by internal PVC segments. A subport set consists of two primary frame handler subports and an optional substitute frame handler subport for each primary.

Substitute Route

In NCP frame relay, an internal PVC segment between a primary frame handler subport and a substitute frame handler subport in a subport set. Contrast with *Primary Route.* See also *Substitute Subport.*

Substitute Subport

In NCP, a frame handler subport in a subport set that is used when a primary frame handler subport in the set is not available.

Subtask

A unit of work whose environment is established by a main task, but has its own TCB, and is displaceable by the operating system.

Subvector

A subcomponent of the NMVT major vector.

SVC

See *Switched Virtual Circuit*.

Switched Virtual Circuit

An X.25 circuit that is dynamically established when needed. the X.25 equivalent of a switched line.

Synchronous Data Link Control (SDLC)

A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the ANSI and High-Level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop.

SYSPARMS

System parameters—values that affect some NetMaster system capabilities. Some SYSPARMS can be modified dynamically.

System Management Facility (SMF)

An optional control feature of OS/390 and z/OS that provides the means for gathering and recording information that can be used to evaluate system usage.

System Services Control Point (SSCP)

A function that VTAM implements to exchange data between CNM applications and PUs in an SNA network

Systems Network Architecture (SNA)

This term describes the logical structure, formats, protocols, and operational sequences for transmitting communication data through the communication system (Fujitsu equivalent is FNA). A set of standards that allows the integration of all the different IBM hardware/software products into a universal network. Introduced in 1974.

TDU

See *Topology Database Update.*

TESP

See *Terminating Equipment Subport*

Terminal Equipment Subport

A subport that serves as a termination point on a virtual circuit.

Terminating Equipment Subport (TESP)

The endpoint of a frame relay permanent virtual circuit; the point at which frame relay terminating equipment has access to the PVC. A terminating equipment subport provides higher level functions with access to a frame relay physical line. Each terminating equipment subport in a single frame relay port has a unique Data Link Connection Identifier (DLCI). Contrast with *Frame Handler Subport.*

TG

See *Transmission Group.*

Thread

A unit of work running under the control of an application program.

Time Sharing Option (TSO)

Allows terminal operators to interact directly with computer resources and facilities. Used mainly by application and system programmers. (Fujitsu equivalent is TSS).

Timestamp

The instant of time at which the information described by a data item was valid.

Token Ring

(1) According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations.
(2) A FDDI or IEEE 802.5 network with a ring topology that passes tokens from one attaching ring station (node) to another. See also *LAN.*

Topology

In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

Topology Database Update (TDU)

A message about a new or changed link or node that is broadcast among APPN network nodes to maintain the network topology database, which is fully replicated in each network node. A TDU contains information that identifies the following:

- The sending node
- The node and link characteristics of various resources in the network

- The sequence number of the most recent update for each of the resources described

Transmission Group (TG)

(1) A connection between adjacent nodes that is identified by a transmission group number.

(2) In a subarea network, a single link or a group of links between adjacent nodes. When a transmission group consists of a group of links, the links are viewed as a single logical link, and the transmission group is called a *multilink transmission group.* A *mixed-media multilink transmission group* is one that contains links of different medium types (for example, token-ring, switched SDLC, nonswitched SDLC, and frame-relay links).

(3) In an APPN network, a single link between adjacent nodes.

TSO

See *Time Sharing Option.*

UAMS

See *Userid Access Maintenance Sub-system*.

UDB

See *User Data Base*.

UDM

See *UnDeliverable Message.*

UnDeliverable Message (UDM)

A term that applies to the Network Partitioning Facility (NPF) of Management Services. It describes a message that cannot be directed to a terminal operator partitioned for the resource to which the message refers, or a message that does not apply to a specific resource.

Unformatted System Services (USS)

A VTAM term that describes a facility that translates an unformatted command such as LOGON or LOGOFF, into a field formatted command for processing by formatted system services. Applies to terminals before connection to an application.

User Data Base (UDB)

(1) UDB file access method layer allowing file access from NCL.

(2) A term used to identify VSAM datasets to which NCL procedures may have access using the &FILE verb (GET, PUT, ADD, and DEL options).

User ID

Defines the function and privilege level to which a specific user is entitled when they sign on to the system. It is associated with a secret password to prevent use by unauthorized personnel. This definition is stored in the UAMS dataset or on an external security system.

Userid Access Maintenance Sub-system (UAMS)

The security component of Management Services that supports the definition of authorized users and their associated function and privilege levels.

USS

See *Unformatted System Services*.

Verb

The term given to a stand-alone statement in an NCL program. NCL *verbs* cause actions to occur. There are different types of verbs, some that dictate the flow of processing and logic, others that fetch information for the procedure to process and others that cause data to flow to external targets.

VFS

See *Virtual File Services*.

Virtual File Services (VFS)

The VSAM dataset, used by many facilities as a database.

Virtual Machine (VM)

A superset operating system that allows other operating systems to run as if they each had their own machine.

Virtual Route (VR)

(1) In SNA, either (a) a logical connection between two subarea nodes that is physically realized as a particular Explicit Route, or (b) a logical connection that is contained wholly within a subarea node for intranode sessions. A VR between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual route pacing, and provides data integrity through sequence numbering of path information units.
(2) Contrast with *Explicit Route.*

Virtual Storage Access Method (VSAM)

A method for processing data files that utilizes relative, sequential, and addressed access techniques. Conceptually identical to ENSCRIBE.

Virtual Telecommunications Access Method (VTAM)

A suite of programs that control communication between terminals and application programs.

Vital Product Data (VPD)

Hexadecimal data used to describe a particular device and its associated software

VM

See *Virtual Machine.*

VM/ESA

An enhanced version of VM that supports 31-bit addressing.

VPD

See *Vital Product Data*.

VR

See *Virtual Route*.

VSAM

See *Virtual Storage Access Method*.

VTAM

See *Virtual Telecommunications Access Method*.

WAN

See *Wide Area Network.*

Wide Area Network (WAN)

(1) A network that provides communication services to a geographical area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities.
(2) A data communication network designed to serve an area of hundreds or thousands of kilometers; for example, public and private packet-switching networks, and national telephone networks.
(3) Contrast with *Local Area Network.*

Wildcard

The term used to describe the character used (usually an asterisk) when defining resources generically—no specific matching character is required in the wildcard character position.

z/OS

An IBM operating system capable of supporting 64-bit architecture.

z/VM

An IBM operating system capable of supporting 64-bit architecture.

Index

Symbols

$NCCNFG procedure 15-37

$NMSMF macro 3-8, E-2, E-8, F-2, F-6, G-1

$NSINIT procedure 6-11, 9-2, 15-37

$NTFMHDF macro H-3

$NTRUDEF macro H-2

$NTSCDEF macro H-2

$NW386SO procedure D-2, D-3

$NWALERT procedure 15-40

$NWCNMPR procedure 2-8

$NWCNMRO procedure 8-6

$NWDS13B procedure D-2, D-4

$NWDSPLY procedure B-11

$NWFCSSO procedure D-2, D-8

$NWLPDA2 procedure D-2, D-9

$NWRTMSO procedure D-2, D-12

$NWRUNCM procedure D-2, D-14

$NWSOLCT procedure D-2, D-16

$NWVPDSO procedure D-2, D-18

&CNMALERT verb 2-8, 2-10, 15-39, 15-40

&CNMCLEAR verb 2-10

&CNMCONT verb 2-10

&CNMDEL verb 2-10

&CNMPARSE verb 2-10

&CNMREAD verb 2-8, 2-10, B-10

&CNMSEND
- statement E-1
- verb 2-10, B-10

&CNMVECTR verb 2-10

&NEWSAUTH verb 2-10

&NEWSRSET verb 2-11

&SNAMS verb 2-4, 2-11

A

ACBs
- CNM 10-2
- MAI 3-10
- NEWS 2-3
- NTS 9-5, 10-2

accounting data 3-5, 3-7, 3-12
- NTS 9-5, 9-7, 9-13, C-7, C-11, C-14
- NTS-SI C-20

activity log 1-8, C-9

administrative tasks
- on startup 6-2

advanced peer-to-peer networking, *see* APPN networking

alert monitor 2-8

ALERTHIST parameter group 15-5

ALERT-NETOP NEWS application E-1

alerts 15-39, B-9
- APPN E-1
- basic 15-42
- CNMPROC procedure 15-39
- creating 15-41
- event filters 6-7
- generic 15-45
- NCL procedures 15-39, 15-40
- NEWS 2-8
- non-generic 15-42
- operators 15-42
- subvectors 15-44, 15-47
- types 15-41

alias names
- and NTS 3-6
- definitions 15-28
- translation facility 2-12, 15-26, E-7

APPN networks 2-4, 2-8, 15-15, B-9

attention messages 15-39
- NTS C-5

attentions, real-time 15-39

Automation Services
- and NetMaster for SNA 1-9

AVAL event type 2-6

B

batch command interface D-4

boundary values, RTM C-2

buffers and trace data 9-18

BYPS event type 2-7

C

class definitions C-15

class-of-service names 15-27

CMC C-17

CNM
- application B-1
- information carrier B-2
- interface
  - concepts B-1
  - NEWS 2-3, 2-8, 15-39, B-8
  - NTS 3-5, 10-2
  - SNA networks 2-3
  - VTAM E-1
- record B-10, C-13
- record processing 6-5, 15-23
- requests, SSCP-related B-7
- user exits E-1

CNM parameter group 6-15, 7-2, 10-2, E-2

CNMFILTERS parameter group 2-7, 6-7

CNMLOGGING parameter group 8-2, 8-5, 8-6

CNMPERFOBJ parameter group 6-8

CNMPROC procedure 2-6, 2-7, 2-8, 6-8, 8-5, 15-39, B-9, B-10, B-11
- record processing 2-8

CNM-RUs B-8

command entry facility 1-8

commands, general
- authority 1-6
- batch command interface D-4
- CAS tables 14-2
- entering commands 1-8
- NetView emulation 1-7
- NEWS commands, summary 2-11
- OS/390 operating system 1-7
- VTAM 1-6, 10-2

commands, specific
- DEFALIAS 15-26, 15-29, 15-30
- DEFCLASS 9-2, C-2, C-3
- DEFCLASS RESOURCE 9-7
- DEFCLASS RTM 9-10
- DEFCLASS SAW 9-8
- DEFCLASS SESSION 9-5
- DELALIAS 15-26, 15-32
- DELCLASS 11-4
- ISR 15-14, C-21
- NTSDBMOD 9-18, 11-6
- NTSMOD 3-7, 11-3
- REPALIAS 15-26, 15-31
- REPCLASS 11-4, C-2
- SAWARE STOP CLOSE 9-16
- SHOW DEFALIAS 15-28
- SHOW DEFCLASS 11-4
- SHOW SKEEP 11-6
- SHOW UDBUSERS 8-5
- STRACE 3-3, 11-5, C-11, C-12
- SYSPARMS 9-11, 9-15, 11-2, 11-5, 11-6, 11-7, C-11
- UDBCTL 3-9
- XLATE 15-33
- Z NET,QUICK 7-3

Communications Management Configuration, *see* CMC

Communications Network Management, *see* CNM

configuration database and NCS 15-37

control
- codes 2-5
- database 2-5, B-8
- database, control codes 2-5

correlation interval 9-17, C-13

CSCF Batch Command Interface D-4

CUST event type 2-6

D

data
- accounting 3-5, 3-7, 3-12, 9-7, 9-13, C-7, C-11
- APPN networks 2-4
- correlation interval 9-17, C-13
- ISR 2-5, 3-5
- MAI 3-6
- network 2-11
- NEWS 2-3, 2-13
- NTS 3-2, 3-4, 9-1, 10-1, C-1, C-16, C-18, C-20
  - session trace 3-3, 3-5
- resources C-10
- response time 2-2, C-13
- route configuration 3-4
- RTM 3-5, 3-11, C-5, C-9, C-12, C-14, C-20
- SAW 3-2, 3-5, 3-11, 10-2, C-9, C-14, C-18
- sessions C-4, C-10, C-14, C-20
- sharing, SAW C-18
- SNAMS 2-4
- solicited 2-2, 2-8, 10-2, B-10, C-13, C-21
- SSCP 2-3, 3-2
- statistical 2-6
- trace 3-12, 9-7, 11-3, C-10, C-11, C-14, C-20
- unattended solicitation 2-11
- unsolicited 2-2, 2-6, 2-8, 10-2, B-11, C-13, C-14, C-21

database
- NEWS 6-6, 6-9, 8-2
- NTS 3-9, 3-12, 9-5, 9-16, 10-1, 10-3, C-14
  - initialization strategy I-3
  - maintenance 11-6
- slots, NTS sessions 3-9

detail records 8-4

device
- configuration 7-3
- solicitation procedures D-2

DEVICESUPP parameter group 6-9

displays, size limits 12-2

DLRC event type 2-6

dormant network C-18

DSECT macro 3-8, F-6, G-1

E

EDS, NTS event generation 3-8, 9-7, 9-17

entry points
- management 15-19, 15-20
- nodes 15-15

ENV event type 2-6

EQUATES CAS table 14-2

event filters 6-7

Event ID B-9

events
- and EDS 3-8
- characteristics 2-7
- filtering 2-7
- generation, NTS 9-17
- NEWS 2-6
- types 2-6, 2-7, 6-7

exits, *see* user exits

F

filtering events 2-7, 6-7

focal points
- backup 15-16
- local 15-16
- management 15-15, 15-17
- nesting 15-16
- nodes 15-15

function codes, NEWS user exits E-5

I

ICS 6-2

ICS parameter groups
- ALERTHIST 15-5
- CNM 6-15, 7-2, 10-2, E-2
- CNMFILTERS 2-7, 6-7
- CNMLOGGING 8-2, 8-5, 8-6
- CNMPERFOBJ 6-8
- DEVICESUPP 6-9
- ISRIN 3-6, 7-2, 9-20, 15-14
- ISROUT 3-6, 9-20, 15-14
- NCS 12-2
- NEWS 6-4, 6-10, 8-5
- NEWSDBOPTS 6-6, 6-10, 7-2
- NSCNTLCACHE 6-6
- NTS 6-4
- PPINETVALRT 6-9, 15-11
- SAW 3-5, 6-15, 9-13, 10-2, 11-7, F-2, H-5
- SAWLOG 10-3
- SMFT37 6-8
- SNAINIT 6-5, 6-11, 6-12, 14-2
- SOCKETS 15-7

IMP event type 2-6

IMR event type 2-6

INFO/MASTER
- and NCS configuration data 15-38

Information Database 1-8

INMC 1-5, 1-6, 3-6, C-15

INST event type 2-6

intensive message recording 9-16

interface, MAI/NTS 3-11, 3-12

Inter-Management Services Connection, *see* INMC

Inter-System Routing, *see* ISR

INTV event type 2-6

ISR
- and MAI 9-20
- and NEWS 2-5, 15-14, 15-39, 15-40
- and NTS 3-6, 3-12, 15-14, C-6, C-8, C-11
- and NTS-SI 3-13, C-14, C-18, C-21
- communication links 9-20, 15-14, E-1
- data 2-5, 3-5
- handshake C-8

ISRIN parameter group 3-6, 7-2, 9-20, 15-14

ISROUT parameter group 3-6, 9-20, 15-14

ISTPDCLU VTAM interface 3-5

L

LAN
- IBM LAN Manager 1-7
- management 1-7

License Manager Program (LMP) keys 1-9

line commands, device solicitation D-2

logmode names 15-27

M

MAI
- and NTS 1-4, 3-6, 3-10, 3-12, 9-20
- data 3-6
- interface to NTS 3-11, 11-7
- session logging 3-12, 9-16

Management Services
- and NetMaster for SNA 1-9
- commands 1-8

masks, resource 2-8

master records 8-4

MSUs B-9
- for NEWS 2-4

Multiple Application Interface 3-10

Multiple Domain Support 2-4

multiple domains, *see* VTAM domains

N

NAUs 3-2

NCL
- procedures
  - NCPView 13-5
  - NCS 15-37
  - NEWS 2-10, 15-39, 15-40, B-8, B-10
  - summary, NEWS 2-10
  - tailoring 6-11
- verbs, summary, NEWS 2-10

NCP 1-5, 5-2
and NCPView 5-2, 13-5
dumps
storage needs 15-35
unformatted 15-34
operations 7-2
parameters 7-2
statistics records 7-2

NCPView
and communications processors 5-1
and NCP 5-2
and SPO 5-2
and VTAM 5-2
exit 13-5
NCP dumps 15-35
overview 1-5
tailoring 13-5

NCS
and INFO/MASTER 15-38
and VTAM 4-2
configuration data 4-2, 15-37
displays, size limits 12-2
graphic displays 4-1, 12-2
overview 1-5
summary displays 4-1, 12-2
tailoring 12-1, 15-39

NCS parameter group 12-2

NetMaster for SNA
components 1-1, 1-2
LMP keys 1-9
overview 1-2
system requirements 1-9

NetSpy and NetMaster for SNA 1-9

NetView operator command emulation 1-7, 6-11

network
addressable units 3-2
alias name translation, *see* alias names translation facility
data 2-11
definitions, NTS 3-6
dormant C-18
management 1-1
reference C-18
services control file 2-5, B-8
see also NEWS control file
shutdown 7-3
statistics 6-8

Network Control Program, *see* NCP

Network Control System, *see* NCS 1-5

Network Error Warning System, *see* NEWS

Network Management Facility 1-6

Network Management Vector Transport, *see* NMVT

Network Services
Control File, *see* NSCNTL
request units, *see* NS RUs

NEWS
and ISR 2-5
and NTS 2-2, C-13
assigning record processing codes B-9
command summary 2-11
control file 2-5, 6-5, 15-23, B-8, B-10
control file maintenance 15-23
data 2-2, 2-13
database 6-6, 6-9, 8-2
device recognition B-9
events 2-6, 6-7
NCL procedures 2-10
NCL verbs summary 2-10
NEWSFILE 8-5
NEWSFILE and CNMPROC 8-5
overview 1-3, 2-1
parameters 8-1
procedures, *see* NEWS NCL procedures
Process ID B-10
processing path selection B-10
record processing 2-5, B-8
record type B-8
SMF exit sample E-8
tailoring 7-1
user exits 2-13, E-1
coding requirements E-3
function codes E-5
initialization E-2
issuing messages E-7
parameter list E-4
processing E-3
registers E-3
samples E-2
verbs, *see* NEWS NCL verbs

NEWS parameter group 6-4, 6-10, 8-5

NEWSDBOPTS parameter group 6-6, 6-10, 7-2

NEWSEXIT exit member E-2

NEWSFILE, *see* NEWS

NEWSXSMF exit member E-2, E-8

NMNTTABS
- compile errors H-4
- modification H-5
- tables H-3
- using H-1

NMVT
- alerts, *see* alerts
- records B-8
- RU 15-41, B-4

NS RUs B-2

NSCNTL and NEWS 2-5, B-8
- *see also* NEWS control file

NSCNTLCACHE parameter group 6-6

NTFY event type 2-6

NTS
- accounting data 9-5, 9-13, C-11, C-14
- and MAI 3-12, 9-16, 11-7
- and NEWS 1-3, 2-2, C-13
- and SMF 11-7
- and SNA 9-20
- and SNI 3-6
- buffer pool definition 3-7
- class definitions 9-2, 11-4, C-1, C-4, C-15
- CNM-RUs B-8
- data 3-2, 3-4, 9-1, 10-1, C-1, C-16, C-18, C-20
  - response time C-13
  - VTAM interfaces 3-5
- database
  - connecting and disconnecting 10-3
  - connecting to 3-9
  - historical information 10-1, I-2
  - initialization strategy I-3
  - logging 3-9, 9-5, 9-7, 9-16, 10-3
  - MAI session logging 3-12
  - maintenance 11-6
  - single image C-14
  - space requirements I-2
- intensive message recording 9-16
- interface to MAI 3-11
- maintaining 11-1
- modifying NTS processing 11-2
- network definitions 3-6
- overview 1-4, 3-1
- processing and storage requirements I-1
- resources, *see* resources
- return codes F-6
- route configuration data 3-4
- RTM data, *see* RTM
- SAS reports 11-7
- session awareness data 3-2
- sessions, *see* sessions
- setting up 6-12
- Single Image C-11
- SMF record formats G-2
- SNA descriptor table H-1
- storage allocation 3-7
- storage requirements I-1
- SYSPARMS command 9-11, 9-15
- system parameters 9-11
- tailoring 9-1, 11-2
- trace data 9-7, 11-5, C-10
- user exits F-1
  - function codes F-4
  - implementing F-1
  - initialization F-2
  - issuing messages F-7
  - MAI sessions 3-12
  - parameter list F-3
  - processing F-2
  - registers F-3
  - resource monitoring C-9
  - resource statistics C-9
  - session data 3-8
  - single image data presentation C-14

NTS parameter group 6-4

NTS-SI 3-12, 3-13, C-11, C-14
- storage requirements I-2

NTSXSMF exit member F-1

O

objective
- percentage C-2
- response time C-2

OCS
- and NTS class definitions 9-2
- and PIUs C-10
- CNMPROC messages 2-8

P

PAFF event type 2-6

parameter groups
  ALERTHIST 15-5
  CNM 6-15, 7-2, 10-2, E-2
  CNMFILTERS 2-7, 6-7
  CNMLOGGING 8-2, 8-5, 8-6
  CNMPERFOBJ 6-8
  DEVICESUPP 6-9
  ISRIN 3-6, 7-2, 9-20, 15-14
  ISROUT 3-6, 9-20, 15-14
  NCS 12-2
  NEWS 6-4, 6-10, 8-5
  NEWSDBOPTS 6-6, 6-10, 7-2
  NSCNTLCACHE 6-6
  NTS 6-4
  PPINETVALRT 6-9, 15-11
  SAW 3-5, 6-15, 9-13, 10-2, 11-7, F-2, H-5
  SAWLOG 10-3
  SMFT37 6-8
  SNAINIT 6-5, 6-11, 6-12, 14-2
  SOCKETS 15-7

path information units C-10

PERF event type 2-6

PERM event type 2-6

PIUs 3-3, 11-5, C-10

PPI interface and NEWS 2-5

PPINETVALRT parameter group 6-9, 15-11

PROC event type 2-6

procedures, device solicitation D-1
  line commands D-2

Process ID in NEWS records B-10

R

RECFMS
  NS RUs B-6
  records B-8, B-9

RECMS
  NS RUs B-7
  records 6-8, B-8, B-9

record processing
  CNMPROC 2-8
  NEWS B-8

reference network C-18

REQMS
  NS RUs B-5
  type 1 B-5
  type 2 B-5
  type 3 B-5
  type 4 B-6
  type 5 B-6
  type 6 B-6

Request Units B-2

Resource ID B-9

resources
  alias name translation 15-26
  class definitions 9-2, 9-5, C-6
  class processing C-6
  data C-10
  masks 2-8
  monitoring 3-2, C-9
  statistics 9-14, C-7, C-8, C-9, G-1
    collection intervals 9-19
    RTM C-8
  trace request C-11

Response Time Monitor C-12

RLST event type 2-7

ROF overview 1-6

route configuration data 3-4

RTM 3-3, C-3
  boundary values C-2
  class definitions 9-2, 9-3, 9-6, C-2, C-4
  class processing C-5
  data 2-2, 3-3, 3-5, 3-11, 6-8, C-2, C-5, C-9, C-12, C-14, C-20
  resource statistics C-8

RUs B-2, H-1
  CNM-RU B-8
  deliver B-3
  forward B-3
  NMVT NS B-4
  RECFMS NS B-6
  RECMS NS B-7
  REQMS NS B-5
  translate-inquiry B-8, E-7
  translate-reply B-8

S

SAW
- class definitions 9-2, 9-3, 9-7, 11-5, C-2, C-4, C-10, C-11
- data 3-2, 3-5, 3-11, 10-2, C-9, C-14, C-18
  - buffers 9-18
  - sharing C-18
- interface concepts C-1
- processing 10-2

SAW parameter group 3-5, 6-15, 9-13, 10-2, 11-7, F-2, H-5

SAWLOG parameter group 10-3

SCUR event type 2-6

secondary program operator 5-2

security 1-6
- exit, full A-1
- structured field descriptions A-2
- UAMS 6-12, A-1

sense codes, SNA 3-8, H-1

session awareness data 3-2, 3-5

Session Replay facility 1-8

sessions
- arrival processing C-15, C-23
- awareness processing 10-1
- class definitions 9-2, 9-3, C-3, C-4
- class processing C-5
- classifications C-4
- data C-4, C-10, C-14, C-20, G-1
  - history I-2
  - sharing C-20, C-22
    - rules C-19
  - storage I-1
- database slots 3-9
- end processing 3-8, C-15, C-23
- events 3-8, 9-17
- keep counts 3-9, 9-7, 9-18, 11-6, I-3
- MAI and SNA 3-10, 9-16, 9-20
- partner names 3-9, 9-5
- partners 3-12, C-14, C-20
- SAW and trace data buffers 9-18
- shutdown processing 9-15
- single image C-14
- start processing 3-7
- storing data for C-10
- trace data 3-3, 3-5, C-14
- virtual MAI 3-10
- warm start 10-2

single image session C-14

SMF
- and NEWS 2-13, E-8
- and NTS 3-8, 3-12, 11-7, C-9, F-2
- database F-1
- record formats E-8, G-2
- record processing 6-8, E-2
- resource statistic processing C-9

SMFT37 parameter group 6-8

SMFWTM macro F-2

SNA
- code translation H-1
- event type 2-6
- Management Services, *see* SNAMS
- MSUs 2-4, B-9
- Network Interconnection 3-6
- resource session status codes 1-5
- sense codes 3-8
- sessions
  - MAI 3-10
  - NTS 9-20

SNAINIT parameter group 6-5, 6-11, 6-12, 14-2

SNAMS 15-15, 15-19
- data 2-4

SNI
- and NEWS 15-27
- and NTS 3-6

SOCKETS parameter group 15-7

SPO and NCPView 5-2

SSCP B-1, C-9
- data 2-3, 3-2

star network and NTS 9-20, C-17

statistical data 2-6

storage requirements I-1

structured field description, security A-2

SYSCMD overview 1-7

System Management Facility C-9

system requirements 1-9

System Services Control Point B-1

T

tailoring
- NCL procedures 6-11
- NCPView 13-5
- NCS 12-1
- NEWS 7-1
- NTS 9-1

TEMP event type 2-6

threshold for network statistics 6-8

trace
- data 3-3, 3-5, 3-12, 9-7, 11-3, 11-5, C-10, C-11, C-14, C-20
  - buffers 9-18
- limits, NTS 9-17, 11-5

translate-inquiry RUs B-8, E-7

translate-reply RUs B-8

U

UAMS security A-1

Unicenter TNG, and alerts 1-3

USER event type 2-6

user exits
- CNM E-1
- defining 6-15
- full security A-1
- function codes E-5
- NCPView 13-5
- NEWS 2-13, 6-15
  - coding requirements E-3
  - maintaining registers E-3
  - parameter list E-4
  - processing exits E-3
- NTS 3-8, 6-15, F-1
  - function codes F-4
  - initialization F-2
  - issuing messages F-7
  - MAI sessions 3-12
  - parameter list F-3
  - processing F-2
  - registers F-3
  - resource monitoring C-9
  - resource statistics C-9
  - sample 3-8, F-1
  - session data 3-8
  - single image data presentation C-14

V

virtual session, MAI 3-10

VSAM
- control interval I-3
- definitions 8-5
- space 8-5

VTAM
- alias name translation 2-12, 15-26, B-8
- and NCPView 5-2
- and NTS 3-11, 10-2
- CNM interface E-1
- commands 1-6, 10-2
- domains 1-5, 1-6, 3-2, 3-3, 3-5, 3-12, 4-2, 5-2, C-10, C-11, C-14, C-15
- trace data C-10, C-11, C-14

W

warm start 10-2

Z

ZNCUX000 exit 13-5